2

IDA PAPER P-1875

ROBUST PREALLOCATED PREFERENTIAL DEFENSE MODEL

Jerome Bracken
James E. Falk
A. J. Allen Tai
(With a contribution by Richard M. Soland)

September 1986

INSTITUTE FOR DEFENSE ANALYSES
1801 N. Beauregard Street, Alexandria, VA 22311

IDA Log No. HQ 85-30357

The work reported in this document was conducted under IDA's Independent Research Program. Its publication does not imply endorsement by the Department of Defense or any other government agency, nor should the contents be construed as reflecting the official position of any government agency.

This paper has been reviewed by IDA to assure that it meets high standards of thoroughness, objectivity, and sound analytical methodology and that the conclusions stem from the methodology.

This document is unclassified and suitable for public release.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

ADA175 216

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY
DD FORM 254 DATED 1 OCTOBER 1983

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

3. DISTRIBUTION/AVAILABILITY OF REPORT
This document is Unclassified and suitable for public release.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
IDA P-1875

5. MONITORING ORGANIZATION REPORT NUMBER (S)

6a. NAME OF PERFORMING ORGANIZATION
Institute for Defense Analyses

6b. OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION
N/A

6b. ADDRESS (CITY, STATE, AND ZIP CODE)
1801 North Beauregard Street
Alexandria, Virginia 22311

7b. ADDRESS (CITY, STATE, AND ZIP CODE)

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
N/A

8b. OFFICE SYMBOL

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
IDA INDEPENDENT RESEARCH

8c. ADDRESS (City, State, and Zip Code)
N/A

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT	PROJECT NO.	TASK NO.	ACCESSION NO. WORK UNIT

11. TITLE (Include Security Classification)
ROBUST PREALLOCATED PREFERENTIAL DEFENSE MODEL

12. PERSONAL AUTHOR(S). Jerome Bracken, James E. Falk And A.J. Allen Tai (with a contribution by Richard M. Soland)

13. TYPE OF REPORT

13b. TIME COVERED FROM TO

14. DATE OF REPORT (Year, Month, Day)
1986, September

15. PAGE COUNT
175

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Ballistic Missile Defense, Preferential Defense, Strategic Defense, Game Theory, Optimization

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Robust Preallocated Preferential Defense Model (RPPDM) treats the problem of protecting a specified number of targets, which may be identical or different, from attack by a number of weapons which may vary within a specified range. The targets are defended by a specified number of interceptors. The attacker is aware of the number of interceptors but not of their allocation to the defense of targets.

The size of the attack is chosen by the attacker from within a specified range. The robust strategies determined in this model do not require the defender to assume an attack size. Rather, the defender chooses a strategy which is good over a wide range of attack sizes, though not necessarily best for any particular attack size. The attacker, knowing that the defender is adopting a a robust strategy, chooses the optimal attack strategy for the number of weapons he chooses to expend. The expected number of survivors is a function of the robust defense strategy and the optimal attack strategy against this robust defense.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
□ UNCLASSIFIED/UNLIMITED ☒ SAME AS REPORT □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL

22b. TELEPHONE (Include Area Code)

22C. OFFICE SYMBOL

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

IDA PAPER P-1875

ROBUST PREALLOCATED PREFERENTIAL DEFENSE MODEL

Jerome Bracken
James E. Falk
A. J. Allen Tai
(With a contribution by Richard M. Soland)

September 1986

A-1

INSTITUTE FOR DEFENSE ANALYSES

IDA Independent Research Program

PREFACE

This study was conducted as part of the Independent Research Program of the Institute for Defense Analyses, under which significant issues of general interest to the defense research community are investigated.

TABLE OF CONTENTS

I. INTRODUCTION

RPPDM is a computer model designed to solve an extended version of the preallocated preferential defense game as presented by Bracken, Brooks and Falk in [1]. The problem revolves around a defender's attempt to protect T targets against A reentry vehicles. The targets may possess different relative values. (In [1] all the targets were assumed to be identical.) We will assume that these valuations are objective and that the defender is unable to hide the values of particular targets from the attacker. The defender has at his disposal D interceptors with which to destroy the RV's. Each interceptor and RV may be assigned to one and only one target. We assume that the attacker knows the total number of interceptors but is unaware of the specific allocation to defend each target.

In [5] and [6] Matheson solved the case, known as the Basic Game, where the attack size A is known to the defender. When we loosen this restriction, however, the solution of the Basic Game no longer suffices, for the defender must now rely on a single strategy, which must be useful against more than one attack size. In [1] a criterion is proposed for a robust defense which most closely approximates the expected outcome of the Basic Game. Robustnesss problems are formulated as linear programs for four alternate scenarios depicting different combinations of behavioral assumptions regarding the attacker and the defender. The present model, RPPDM, solves the most interesting of these scenarios: the defender believes that the attacker can discover and thus optimize whatever defense he chooses to employ, and the defender is correct (case II,II of [1]).

As it is currently designed, RPPDM may be used in either batch or interactive mode. In interactive mode, RPPDM prompts the user for every piece of information necessary to execute the program. In batch mode a separate program is run to create an input file for use with RPPDM. The user has a choice of sending the report to a file, the terminal, or both. The user may choose to solve any number of robust defenses for a given set of Basic Game parameters. (This will be made clear in the examples.) The output report presents a summary of the game parameters, the optimal strategies for the Basic Game, the expected outcome of the Basic Game, the robust defense, the optimal attacks against the robust defense, and the resulting expected target-survival rates.

The code is organized around a main program, MAIN, whose major function is to query the user (or read from a file) for specific parameters needed to set up the problem, such as the number of interceptors, the number of targets, etc. MAIN then uses that information to call on a series of subroutines to conduct the appropriate numerical caluclations and to generate the solution reports.

The subroutines are organized into three categories:

1) PIJ subroutines
2) LP subroutines
3) REPORT subroutines

The PIJ subroutines include SIMAT1, SIMAT2, SEQAT1, and SEQAT2. They are used to generate the PIJ's associated with particular combinations of attack and defense methodologies. Only one of these subroutines is used in any one run of RPPD.

The LP subroutines include BG, YROUBUST and XROBUST. These subroutines are the heart of the model. They set up the linear programming equivalents of the problem in a format that can be accepted by XMP, a linear programming system. BG is responsible for returning the optimal attacker and defender strategies the expected outcome (game value) of the Basic Game. YROBUST calculates the robust defense. XROBUST solves for the optimal attacks against the robust defense and the resulting expected survival rates.

The REPORT subroutines include SUMMARY, STPRINT, YPRINT, VPRINT, ALPRINT, ALYRPRINT, ALVPRINT, and RVINTCOUNT. They are used to produce the output reports.

The code is written in FORTRAN-77 to run on a VAX 8600 computer.

Note that computer programs which solve the Basic Game are documented in [3], [4], and [7].

II. MAIN PROGRAM

Program MAIN acts as a coordinator between the user and the subroutines where the actual "work" takes place. MAIN first asks the user whether interactive or batch mode is to be used. If batch was chosen it asks fo. the name of the input file. The flow of MAIN may then be broken roughly into five stages:

1) Generate the PIJ's
2) Solve the Basic Game

3) Produce the solution report for the Basic Game
4) Solve the robust game
5) Produce the solution report for the robust game

During each stage, MAIN will prompt the user (or read from file in batch mode) for the additional information needed to process that stage. We will discuss each of these stages separately, focusing on the needed inputs and the subroutines called.

Stage 1: Generate the PIJ's

RPPDM is equipped to handle five combinations of attack and defense methodologies:

A) Simultaneous Attack with One Shot
B) Simultaneous Attack with Shoot-Look-Shoot
C) Sequential Attack of Unknown Size with One Shot
D) Sequential Attack of Unknown Size with Shoot-Look-Shoot
E) Sequential Attack of Known Size with Shoot-Look-Shoot

Depending upon the answers to two questions (three if the attack is sequential), MAIN will call on one of four subroutines, SIMATI1 for A, SIMAT2 for B, SEQAT1 for C and D, and SEQAT2 for E, to generate the appropriate set of PIJ's. For each of the PIJ subroutines the following inputs are necssary: the maximum number of RVs that may defend a single target (S), the failure probability of the RV's (PFA) and that of the interceptors (PFD). For cases involving shoot look shoot defenses, two failure probabilities for the interceptors are necessary (PFD1 and PFD2 rather than PFD).

Stage 2: Solve the Basic Game

Main prompts for the minimum attack size (MINRV), the maximum attack size (MAXRV), and the attack size increment (INCRV). These values are then used to calculate the actual set of attack sizes (RV) and the number of attack sizes (A). The number of interceptors (INT) and the total number of targets (TARGETS) are also needed. In addition, if there are more than one type of target, the relative value VTYPE) and the number of targets (NTAR) belonging to each type must also be entered. The value and the size of each target type is then calculated as the fraction of all the targets. Subroutine BG is called to solve the Basic Game. BG returns the attracker's minimax strategy in XBG, the defender's minimax strategies in YBG, and the game values in VBG.

Stage 3: Produce the solution report for the Basic Game

MAIN calls on the REPORT subroutines to print out the results of the Basic Game on the selected output device(s). SUMMARY provides a listing of the problem's parameters as entered. STPRINT prints out XBG and YBG, VPRINT prints out VBG, ALPRINT prints out allocation tables based on XBG and YBG, ALVPRINT prints out the expected number of targets that will survive for each target type, and RVINTCOUNT prints out the numbers RV's and interceptors assigned to each target type. (See Section VII, "Notes on Output" for more details.)

Stage 4: Solve the Robust Game

MAIN prompts for the number of robust strategies desired. Stages 4 and 5 are repeated until all of them have been solved.

MAIN prompts for a lower and an upper bound on the attack sizes for which a robust defense is to be found. Both bounds must lie in the set of attack sizes defined for the Basic Game during stage 2. Subroutine YROBUST is then called to solve for the robust defense (YII). With the robust defense defined, MAIN calls XROBUST to find the optimal attacks (XII) against YII and the resulting expected survival rates (VII).

Stage 5: Produce the solution report for the Robust Game

MAIN calls on YPRINT to print out YII, STPRINT to print out XII, VPRINT to print out VII. ALYPRINT to print out the allocation table for YII, ALPRINT to print out the allocation table for XII, ALVPRINT to print out the expected number of targets that will survive for each target type, and RVINTCOUNT to print out the numbers of RV's and interceptors assigned to each target type.

III. PIJ SUBROUTINES

The PIJ subroutines SIMAT, SIMAT2, SEQAT1, and SEQAT2 are called on to generate the appropriate set of PIJ's associated with particular attack and defense methodologies.

SIMAT1 is designed to generate the PIJ's when the attack is simultaneous and the defense has only one opportunity to intercept the RV's.

SIMAT2 applies when the attack is still simultaneous, but the defense has two opportunities to intercept the RV.

SEQAT1 applies when the attack is sequential of unknown size and the defender has one or two chances to intercept the RV's. The first case is accomplished by passing 1.0 for the failure rate of the first interceptor salve (PFD1) and assigning PFD to PFD2. In both cases the defender does not know the number of RV's that will arrive at a particular target.

SEQAT2 applies when the attack is sequential, the defender has shoot-look-shoot, and the defender knows (after the attack begins) how many RV's will engage him at a given target.

The resulting PIJ's are returned to MAIN in the two dimensional array P. The first index of P is the number of RV's that attacks a target, and the second index is the number of interceptors that defends a target. Thus P(I,J) is the probability that a target will survive when attacked by I RV's and defended by J interceptors. See Appendix H for discussions of the concepts on which these PIJ subroutines are based.

IV. LP SUBROUTINES

The purpose of BG, YROBUST, and XROBUST is to set up the appropriate linear programs for processing by XMP. In the following subsections we will examine each individually in terms of the linear programs they solve. Please consult program listing in the Appendix and XMP documentation for details regarding the mechanics of the interactions between BG, etc., and XMP. In the linear programs that follow, all variables are, or can be assumed to be, non-negative.

BG

BG is designed to solve the Basic Game. A linear program of the Basic Game may be set up as follows (see Appendix G for the derivation):

$$\mathrm{VBG} = \max_{Y(k,j),s(k),t} \left[\sum_{k=1}^{\mathrm{NTYPE}} s(k)\text{-}(\mathrm{RV/TARGETS}) \cdot t \right]$$

subject to

$$\left.\begin{array}{ll}
s(k) - \mathrm{VF}(k) \displaystyle\sum_{j=0}^{S} \left[P(0,j) \cdot Y(k,j) \right] & \leq 0 \\
s(k) - \mathrm{NF}(k) \cdot t - \mathrm{VF}(k) \displaystyle\sum_{j=0}^{S} \left[P(1,j) \cdot Y(k,j) \right] & \leq 0 \\
\vdots & \\
s(k) - \mathrm{NF}(k) \cdot R \cdot t - \mathrm{VF}(k) \displaystyle\sum_{j=0}^{S} \left[P(R,j) \cdot Y(k,j) \right] & \leq 0
\end{array}\right\} \quad 1 \leq k \leq \mathrm{NTYPE}$$

$$\sum_{j=0}^{S} Y(k,j) = 1 \qquad 1 \leq k \leq \mathrm{NTYPE}$$

$$\sum_{k=1}^{\mathrm{NTYPE}} \sum_{j=0}^{S} \left[j \cdot \mathrm{NF}(k) \cdot Y(k,j) \right] = \mathrm{INT/TARGETS}$$

The problem is solved first for RV(1), the smallest attack size, and the solutions stored in VBG, XBG (the dual variables of the inequality constraints) and YBG. Then RV(1) is substituted by RV(2) and the problem reoptimized. BG keeps substituting RV's until all of the attack sizes are solved.

XBG and YBG are passed back to MAIN as three dimensional arrays. The first index designates the attack size, the second index is the target type, and the third index is the number of RV's (for XBG) or interceptors (for YBG). XBG(A,K,I) and YBG(A,K,J), are, respectively, the fraction of type K targets attacked by I RV's and the fraction of type K targets defended by J interceptors for the basic game with :he A'th attack size (RV(A)).

VBG is passed back to MAIN as a one dimensional array with the index designating the attack size. VBG(A) is the game value for the A'th attack size.

YROBUST

YROBUST is designed to solve for the robust defense. A linear program of the robust game may be set up as follows (see Appendix G for the derivation):

$$\max_{Y(k,j),s(A,k),t(A)} \rho$$

subject to

$$\left.\begin{array}{l} VF(k) \displaystyle\sum_{j=0}^{S} \left[P(0,j)\cdot Y(k,j) \right] - Z(k,0) = 0 \\ \qquad\qquad\qquad\qquad\qquad\qquad\vdots \\ VF(k) \displaystyle\sum_{j=0}^{S} \left[P(R,j)\cdot Y(k,j) \right] - Z(k,R) = 0 \end{array}\right\} \qquad 1 \le k \le NTYPE$$

$$\rho \cdot VBG(A) - \sum_{k=1}^{NTYPE} s(A,k) + \left[RV(A)/TARGETS \right] \cdot t(A) \quad \le 0 \qquad 1 \le A \le \overline{A}$$

$$\left.\begin{array}{ll} s(A,k) - Z(k,0) & \le 0 \\ s(A,k) - Z(k,1) - NF(k)\cdot t(A) & \le 0 \\ \qquad\vdots & \\ s(A,k) - Z(k,R) - NF(k) \cdot R \cdot t(A) & \le 0 \end{array}\right\} \qquad \begin{array}{l} 1 \le A \le \overline{A} \\ \\ 1 \le k \le NTYPE \end{array}$$

$$\sum_{j=0}^{S} Y(k,j) = 1 \qquad 1 \le k \le NTYPE$$

$$\sum_{k=1}^{NTYPE} \sum_{j=0}^{S} \left[j \cdot NF(k) \cdot Y(k,j) \right] = INT/TARGETS$$

The Y that solves this program is stored in YII and passed back to MAIN as a two dimensional variable. The first index is target type, and the second index is the number of interceptors assigned to a target. YII(K,J) is the fraction of type K targets defended by J interceptors under the robust defense.

XROBUST

Given the robust defense, YII, XROBUST finds the optimal attack, XII, and the resulting expected survival rate using the following LP:

$$\mathrm{VII} = \min_{X(k,i)} \sum_{k=1}^{\mathrm{NTYPE}} \mathrm{VF}(k) \sum_{i=0}^{R} \left\{ \left[\sum_{j=0}^{S} P(i,j)\ \mathrm{YII}(k,j) \right] X(k,i) \right\}$$

subject to

$$\sum_{i=0}^{R} X(k,i) = 1 \qquad 1 \le k \le \mathrm{NTYPE}$$

$$\sum_{k=1}^{\mathrm{NTYPE}} \sum_{i=0}^{R} \left[i \cdot \mathrm{NF}(k)\ \cdot X(k,i) \right] = \mathrm{RV/TARGETS}$$

After the LP is solved, and the X that yields VII is stored in XII for every attack size.

XII and VII are passed back to MAIN as, respectively, two and one dimensional variables. The indices are the same as their counterparts in the Basic Game, XBG and VBG.

V. REPORT SUBROUTINES

The report subroutines generate formatted output for RPPDM. The SUMMARY subroutine creates a copy of the parameters of the game as specified and send it to the output device selected. The STPRINT subroutine prints out attacker and defender strategies that are indexed by attack sizes (i. e., XBG, YBG, and XII). The YPRINT subroutine prints out the robust defense. The VPRINT subroutine prints out the expected survival rate for the chosen range of attack sizes. ALPRINT prints out target allocation tables for attacker and defender strategies indexed by attack sizes. ALYPRINT prints out target allocation tables for the robust defense. ALVPRINT prints out the expected numbers of targets that will survive for each target type. RVINTCOUNT prints out the missile (RV or interceptor) allocation by target types.

VI. NOTES ON INPUT

A separate program, BATCH, is provided to facilitate the use of batch mode. It is an interactive program that creates an input file that may be read by RPPDM. It will query the user for every piece of information that is needed to run RPPDM (in the same order as the interactive mode), and write the information to a sequential file named by the user. Thus when a user seeks to run a series of problems with changes in only a small number of the parameters, he or she could create the first problem using BATCH, then use an editor to modify the input file for the first problem to fit the data for the other problems. We demonstrate this procedure in the seccnd example of Section VIII.

VII. NOTES ON OUTPUT

In any given run of RPPDM a minimum of 9 tables are printed, and for each robust defense desired, an additional 8 tables are added. (For runs involving targets of equal value, the numbers are 7 and 6, since the RV and interceptor allocation tables are eliminated.)

The first table is a table of the basic parameters that define the Basic and the Robust Games.

The next eight tables list the results of the Basic Game. The first two of these tables print out, respectively, the attacker and defender strategies indexed by target types and then by attack sizes. For each attack size A, there are N entries which define the minimax strategy for a given target type K. The i'th entry corresponds to the fraction of type K targets that will be assigned i-1 RV's or interceptors. Each row contains a maximum of 10 entries. The first row will include the first 10 entries, the second row the next 10 entries, etc. Therefore, an entry in the third row and sixth column is the fraction of targets (of that target type) to receive 25 RV's or interceptors.

The third table of the Basic Game is the game value, or the expected survival rate associated with the minimax strategies of the Basic Game. Each entry is the fraction of the "total value" that is expected to survive. Thus .3567 means that the expected value of all the targets that survive the attack is 35.7% of the original value before the attack.

The fourth and the fifth tables are just a variation on the strategy tables discussed earlier. These target allocation tables, instead of using fractions, give the actual numbers that correspond to the minimax strategies. The allocation tables are first subdivided by attack sizes and then by target types. For each attack size the strategy for all the different target types are listed together. The Kth entry now corresponds to the number of targets (of the target type under consideration) that should be assigned K-1 missiles.

The seventh table lists the number of targets of each target type that is expected to survive in the Basic Game.

The eighth and ninth tables lists the RV and interceptor allocations by target type. Each entry corresponds to the total number of missiles assigned to all the targets of that target type. (As noted above, these tables are not created when there is only one type of target.)

The next eight tables correspond to the first Robust Game (if any). The tables are printed out and the entries defined in an identical way to those in the Basic Game except that the defender's strategy and target allocation tables are not indexed by attack sizes. The first table lists the robust defender's strategy, the second the optimal attacker strategies against the robust defense, the third the expected value of teh robust game, the fourth and fifth the target allocation tables for the defender and the attacker (respectively), the sixth the expected number of targets surviving, by target type, and the seventh and eighth the missile allocation tables for the attacker and the defender in this, the first Robust Game.

If another robust defense is called for, then eight more tables will be printed in the same order as the eight associated with the first Robust Game.

VIII. HOW TO USE RPPDM

The following is a list of the FORTRAN files in RPPDM and their contents:

File	Contents
RPPDM.FOR	MAIN

PIJ1.FOR	SIMAT1
PIJ2.FOR	SIMAT2
PIJ3.FOR	SEQAT1
PIJ4.FOR	SEQAT2
LP1.FOR	BG
LP2.FOR	YROBUST
LP3.FOR	XROBUST
REPORT1.FOR	SUMMARY,STPRINT
	YPRINT, VPRINT
REPORT2.FOR	ALYPRINT
	ALPRINT, ALVPRINT
REOIRT3.FOR	RVINTCOUNT
BATCH.FOR	BATCH

In addition, the XMP linear programming package is necessary for the use of RPPDM.

Together with the XMP subroutines, all the files except BATH.FOR must be compiled and then linked into an executable image. BATCH.FOR should be compiled and linked separately.

The following examples illustrate typical runs of the model. In the first example the user wishes to examine the robust defense under simultaneous attack and one shot defense with only one type of target. In the second example the user wants to find two different robust defenses under a more complicated combination of attack and defense methodologies with 3 different types of targets. In the first case he or she interactively provides the model with all the specifications needed. In the second case he or she will create an input file with BATCH and then use batch mode with RPPDM.

In order to solve the Basic Game, RPPDM needs the attack and defense methodologies, the RV and interceptor failure rates, the minimum and maximum attack sizes, the attack size increment, the number of interceptors, the total number of targets, the relative value of each target type, and the number of targets in each target type. For the Robust Game, RPPDM needs to know the minimum and maximum attack sizes in the range of attack sizes for which the robust defense is to be found; both must be in the set of attack sizes for the Basic Game. The same increment from the Basic Game is used here.

These sample runs are shown exactly as they would have appeared on a terminal screen except for two items: the '*****USER SAYS >' segment that preceed every user entry and the comments outlined by slashes. They are provided to highlight and clarify certain situations.

A. EXAMPLE 1

```
$ LINK RPPDM, PIJ/LIB, LP/LIB, XMP/LIB, REP/LIB
$ RUN RPPDM
     ///////////////////////////////////////////(/////////////////////////
     /                                                                   /
     /  On the VAX/VMS operating system there is a library function      /
     /  which allows object files to be placed together in a library     /
     /  file.  Because the PIJ, LP, REPORT, and XMP subroutines were     /
     /  all stored in several different files, library files were       /
     /  created to avoid confusion.                                      /
     /                                                                   /
     /////////////////////////////////////////////////////////////////////

THERE ARE TWO INPUT OPTIONS:
     1) TERMINAL (INTERACTIVE)
     2) FILE     (BATCH)

PLEASE ENTER THE NUMBER OF THE DESIRED OPTION
*****USER SAYS>  1
```

```
You have three options fo  the output of the results :

     1) TERMINAL only
     2) FILE only
     3) TERMINAL and FILE

Please enter the number for the desired option ?
*****USER SAYS>  3

Please type in the desired file name (of less than 10 characters
  including the extension) ?
*****USER SAYS>  TEST1.OUT

     ////////////////////////////////////////////////////////////////
     /                                                              /
     /  A file named TEST1.OUT will be created to hold a copy of the /
     /  output report.  If an extension had not been specified, .DAT /
     /  would be added by default.                                  /
     /                                                              /
     ////////////////////////////////////////////////////////////////

The MAXIMUM number of RV's (up to 30) at a single target ?
*****USER SAYS>  10

The MAXIMUM number of INTERCEPTORS (up to 30) at a single target ?
*****USER SAYS>  10

Select one of the following attack methodologies:

     1) SIMULTANEOUS ATTACK
     2) SEQUENTIAL ATTACK

Please input the number of the desired attack ?
*****USER SAYS>  1

The FAILURE rate of the RV's ?
*****USER SAYS>  .3

Select one of the following defense methodologies:

     1) ONE SHOT
     2) SHOOT LOOK SHOOT

Please input the number for the desired option ?
*****USER SAYS>  1

The FAILURE rate of the interceptors ?
*****USER SAYS>  .3

     ////////////////////////////////////////////////////////////////
     /                                                              /
     /  All the information needed to generate the Pij's are passed /
     /  to SIMAT1, since SIMAT1 corresponds with the attack-defense /
     /  combination the USER has selected.  If another combination  /
     /  had been selected, MAIN may ask for additional information. /
     /  The second example illustrates this case.                   /
     /                                                              /
     ////////////////////////////////////////////////////////////////
```

```
The MINIMUM attack size ?
•••••USER SAYS>  1000

The MAXIMUM attack size ?
•••••USER SAYS>  10000

The attack size INCREMENT ?
•••••USER SAYS>  1000

The NUMBER of interceptors ?
•••••USER SAYS>  6000

The TOTAL NUMBER of targets ?
•••••USER SAYS>  1000

The number of TYPES of targets ?
•••••USER SAYS>  1

     //////////////////////////////////////////////////////////////////
     /                                                                /
     /  All the information needed to solve the Basic Game is passed  /
     /  to BG, which will call on the XMP subroutines to solve the    /
     /  linear program.                                               /
     /                                                                /
     //////////////////////////////////////////////////////////////////

XMAPS...WORDS OF MEMORY AVAILABLE
INTEGER:      10000         REAL:     10000

XMAPS...YOU HAVE ROOM FOR   2987 NON-ZEROS IN THE BASIS FACTORS
YOU COULD REDUCE REAL MEMORY FROM     10000 TO      6509

     //////////////////////////////////////////////////////////////////
     /                                                                /
     /    The above messages are sent by XMP. Ignore them.  If you    /
     /    selected an output device other than the terminal, the     /
     /    messages will not appear there.                             /
     /                                                                /
     //////////////////////////////////////////////////////////////////
```

```
      //////////////////////////////////////////////////////////////
      /                                                            /
      /           SOLUTION REPORT FOR THE BASIC GAME               /
      /                                                            /
      //////////////////////////////////////////////////////////////

THE PARAMETERS OF THIS PREALLOCATED PREFERENTIAL DEFENSE GAME

**********************************************************************

 THE ATTACK METHODOLOGY                           SIMULTANEOUS
 THE DEFENSE METHODOLOGY                          ONE SHOT

 THE FAILURE RATE OF RV'S                         0.300
 THE FAILURE RATE OF THE INTERCEPTORS             0.300

 MAXIMUM NUMBER OF RV'S ATTACKING A
      SINGLE TARGET                                  10
 MAXIMUM NUMBER OF INTERCPTORS DEFENDING A
      SINGLE TARGET                                  10

 THE MINIMUM NUMBER OF RV'S                        1000
 THE MAXIMUM NUMBER OF RV'S                       10000
 THE ATTACK SIZE INCREMENT                         1000
 THE NUMBER OF INTERCEPTORS                        6000
 THE TOTAL NUMBER OF TARGETS                       1000

 THE NUMBER OF TARGET TYPES                           1
       TARGET TYPE  1:
            NUMBER OF TARGETS                      1000
            RELATIVE VALUE                        1.000

**********************************************************************
```

```
THE ATTACKER'S BASIC GAME MINIMAX STRATEGIES

••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••

                  TARGET TYPE    1:  100.00% OF TOTAL TARGETS, WITH 100.00% OF TOTAL VALUE
ATTACK SIZE          0        1        2        3        4        5        6        7        8        9
-----------------------------------------------------------------------------------------------------------------------
    1000         : 0.8138 : 0.0169 : 0.0081 : 0.0000 : 0.0000 : 0.0000 : 0.1611 : 0.0000 : 0.0000 : 0.0000 :
                 : 0.0000 :
               ------------------------------------------------------------------------------------------------------
    2000         : 0.6277 : 0.0338 : 0.0162 : 0.0000 : 0.0000 : 0.0000 : 0.3223 : 0.0000 : 0.0000 : 0.0000 :
                 : 0.0000 :
               ------------------------------------------------------------------------------------------------------
    3000         : 0.4415 : 0.0506 : 0.0244 : 0.0000 : 0.0000 : 0.0000 : 0.4834 : 0.0000 : 0.0000 : 0.0000 :
                 : 0.0000 :
               ------------------------------------------------------------------------------------------------------
    4000         : 0.2554 : 0.0675 : 0.0325 : 0.0000 : 0.0000 : 0.0000 : 0.6446 : 0.0000 : 0.0000 : 0.0000 :
                 : 0.0000 :
               ------------------------------------------------------------------------------------------------------
    5000         : 0.0692 : 0.0844 : 0.0406 : 0.0000 : 0.0000 : 0.0000 : 0.8057 : 0.0060 : 0.0000 : 0.0000 :
                 : 0.0000 :
               ------------------------------------------------------------------------------------------------------
    6000         : 0.0000 : 0.0641 : 0.0736 : 0.0000 : 0.0000 : 0.0000 : 0.2475 : 0.6148 : 0.0000 : 0.0000 :
                 : 0.0000 :
               ------------------------------------------------------------------------------------------------------
    7000         : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 :
                 : 0.0000 :
               ------------------------------------------------------------------------------------------------------
    8000         : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 :
                 : 0.0000 :
               ------------------------------------------------------------------------------------------------------
    9000         : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 1.0000 :
                 : 0.0000 :
               ------------------------------------------------------------------------------------------------------
   10000         : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                 : 1.0000 :
-----------------------------------------------------------------------------------------------------------------------


••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
```

THE DEFENDER'S BASIC GAME MINIMAX STRATEGIES

```
*****************************************************************************************************

                 TARGET TYPE    1:  100.00% OF TOTAL TARGETS, WITH 100.00% OF TOTAL VALUE
ATTACK SIZE         0        1        2        3        4        5        6        7        8        9
-----------------------------------------------------------------------------------------------------
     1000      : 0.1306 : 0.1462 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.3446 : 0.0000 : 0.0000 : 0.0000 :
               : 0.3786 :
               ----------------------------------------------------------------------------------------
     2000      : 0.1306 : 0.1462 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.3446 : 0.0000 : 0.0000 : 0.0000 :
               : 0.3786 :
               ----------------------------------------------------------------------------------------
     3000      : 0.1306 : 0.1462 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.3446 : 0.0000 : 0.0000 : 0.0000 :
               : 0.3786 :
               ----------------------------------------------------------------------------------------
     4000      : 0.1306 : 0.1462 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.3446 : 0.0000 : 0.0000 : 0.0000 :
               : 0.3786 :
               ----------------------------------------------------------------------------------------
     5000      : 0.1306 : 0.1462 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.3446 : 0.0000 : 0.0000 : 0.0000 :
               : 0.3786 :
               ----------------------------------------------------------------------------------------
     6000      : 0.3398 : 0.0343 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0979 : 0.0000 : 0.0000 :
               : 0.5280 :
               ----------------------------------------------------------------------------------------
     7000      : 0.4000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.6000 :
               ----------------------------------------------------------------------------------------
     8000      : 0.4000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.6000 :
               ----------------------------------------------------------------------------------------
     9000      : 0.4000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.6000 :
               ----------------------------------------------------------------------------------------
    10000      : 0.4000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.6000 :
-----------------------------------------------------------------------------------------------------


*****************************************************************************************************
```

```
THE GAME VALUES

******************************

 ATTACK SIZE
______________________________
     1000     :     0.8777
     2000     :     0.7554
     3000     :     0.6331
     4000     :     0.5109
     5000     :     0.3886
     6000     :     0.2826
     7000     :     0.1923
     8000     :     0.1281
     9000     :     0.0853
    10000     :     0.0568
______________________________

******************************
```

THE ATTACKER'S BASIC GAME TARGET ALLOCATION

```
*******************************************************************************************************

                                               ATTACK SIZE =  1000
TARGET TYPE       0         1         2         3         4         5         6         7         8         9
--------------------------------------------------------------------------------------------------------------
     1      :  813.8 :   16.9 :    8.1 :    0.0 :    0.0 :    0.0 :  161.1 :    0.0 :    0.0 :    0.0 :
            :    0.0 :
--------------------------------------------------------------------------------------------------------------

                                               ATTACK SIZE =  2000
                  0         1         2         3         4         5         6         7         8         9
--------------------------------------------------------------------------------------------------------------
     1      :  627.7 :   33.8 :   16.2 :    0.0 :    0.0 :    0.0 :  322.3 :    0.0 :    0.0 :    0.0 :
            :    0.0 :
--------------------------------------------------------------------------------------------------------------

                                               ATTACK SIZE =  3000
                  0         1         2         3         4         5         6         7         8         9
--------------------------------------------------------------------------------------------------------------
     1      :  441.5 :   50.6 :   24.4 :    0.0 :    0.0 :    0.0 :  483.4 :    0.0 :    0.0 :    0.0 :
            :    0.0 :
--------------------------------------------------------------------------------------------------------------

                                               ATTACK SIZE =  4000
                  0         1         2         3         4         5         6         7         8         9
--------------------------------------------------------------------------------------------------------------
     1      :  255.4 :   67.5 :   32.5 :    0.0 :    0.0 :    0.0 :  644.6 :    0.0 :    0.0 :    0.0 :
            :    0.0 :
--------------------------------------------------------------------------------------------------------------

                                               ATTACK SIZE =  5000
                  0         1         2         3         4         5         6         7         8         9
--------------------------------------------------------------------------------------------------------------
     1      :   69.2 :   84.4 :   40.6 :    0.0 :    0.0 :    0.0 :  805.7 :    0.0 :    0.0 :    0.0 :
            :    0.0 :
--------------------------------------------------------------------------------------------------------------

                                               ATTACK SIZE =  6000
                  0         1         2         3         4         5         6         7         8         9
--------------------------------------------------------------------------------------------------------------
     1      :    0.0 :   64.1 :   73.6 :    0.0 :    0.0 :    0.0 :  247.5 :  614.8 :    0.0 :    0.0 :
            :    0.0 :
--------------------------------------------------------------------------------------------------------------

                                               ATTACK SIZE =  7000
                  0         1         2         3         4         5         6         7         8         9
--------------------------------------------------------------------------------------------------------------
     1      :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 : 1000.0 :    0.0 :    0.0 :
            :    0.0 :
--------------------------------------------------------------------------------------------------------------
```

```
                                                 ATTACK SIZE =  8000
                  0         1         2         3         4         5         6         7         8         9

      1      :    0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :  1000.0 :     0.0 :
             :    0.0 :


                                                 ATTACK SIZE =  9000
                  0         1         2         3         4         5         6         7         8         9

      1      :    0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :  1000.0 :
             :    0.0 :


                                                 ATTACK SIZE = 10000
                  0         1         2         3         4         5         6         7         8         9

      1      :    0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
             : 1000.0 :
```

```
THE DEFENDER'S BASIC GAME TARGET ALLOCATION

************************************************************************************************

                                                ATTACK SIZE =  1000
TARGET TYPE        0         1         2         3         4         5         6         7         8         9
----------------------------------------------------------------------------------------------------------------
     1        :  130.6 :  146.2 :    0.0 :    0.0 :    0.0 :    0.0 :  344.6 :    0.0 :    0.0 :    0.0 :
              :  378.6 :
----------------------------------------------------------------------------------------------------------------


                                                ATTACK SIZE =  2000
                   0         1         2         3         4         5         6         7         8         9
----------------------------------------------------------------------------------------------------------------
     1        :  130.6 :  146.2 :    0.0 :    0.0 :    0.0 :    0.0 :  344.6 :    0.0 :    0.0 :    0.0 :
              :  378.6 :
----------------------------------------------------------------------------------------------------------------


                                                ATTACK SIZE =  3000
                   0         1         2         3         4         5         6         7         8         9
----------------------------------------------------------------------------------------------------------------
     1        :  130.6 :  146.2 :    0.0 :    0.0 :    0.0 :    0.0 :  344.6 :    0.0 :    0.0 :    0.0 :
              :  378.6 :
----------------------------------------------------------------------------------------------------------------


                                                ATTACK SIZE =  4000
                   0         1         2         3         4         5         6         7         8         9
----------------------------------------------------------------------------------------------------------------
     1        :  130.6 :  146.2 :    0.0 :    0.0 :    0.0 :    0.0 :  344.6 :    0.0 :    0.0 :    0.0 :
              :  378.6 :
----------------------------------------------------------------------------------------------------------------


                                                ATTACK SIZE =  5000
                   0         1         2         3         4         5         6         7         8         9
----------------------------------------------------------------------------------------------------------------
     1        :  130.6 :  146.2 :    0.0 :    0.0 :    0.0 :    0.0 :  344.6 :    0.0 :    0.0 :    0.0 :
              :  378.6 :
----------------------------------------------------------------------------------------------------------------


                                                ATTACK SIZE =  6000
                   0         1         2         3         4         5         6         7         8         9
----------------------------------------------------------------------------------------------------------------
     1        :  339.8 :   34.3 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :   97.9 :    0.0 :    0.0 :
              :  528.0 :
----------------------------------------------------------------------------------------------------------------


                                                ATTACK SIZE =  7000
                   0         1         2         3         4         5         6         7         8         9
----------------------------------------------------------------------------------------------------------------
     1        :  400.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
              :  600.0 :
----------------------------------------------------------------------------------------------------------------
```

```
                                        ATTACK SIZE =  8000
                 0         1         2         3         4         5         6         7         8         9
--------------------------------------------------------------------------------------------------------------
    1       :  400.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
            :  600.0 :
--------------------------------------------------------------------------------------------------------------

                                        ATTACK SIZE =  9000
                 0         1         2         3         4         5         6         7         8         9
--------------------------------------------------------------------------------------------------------------
    1       :  400.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
            :  600.0 :
--------------------------------------------------------------------------------------------------------------

                                        ATTACK SIZE = 10000
                 0         1         2         3         4         5         6         7         8         9
--------------------------------------------------------------------------------------------------------------
    1       :  400.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
            :  600.0 :
--------------------------------------------------------------------------------------------------------------

**************************************************************************************************************
```

```
THE EXPECTED NUMBER OF TARGETS SURVIVING

**************************

                TARGET TYPE
 ATTACK SIZE  :      1
--------------------------
    1000      :   877.71
    2000      :   755.43
    3000      :   633.14
    4000      :   510.86
    5000      :   388.57
    6000      :   282.57
    7000      :   192.34
    8000      :   128.08
    9000      :    85.30
   10000      :    56.81
--------------------------

**************************
```

```
Please enter the number of different ranges of RV's for which robust
  solutions are to be found ?
        ***** Enter 0 if no robust solution is desired*****

     //////////////////////////////////////////////////////////////////////
     /                                                                    /
     /            If '0' were entered, the program would                  /
     /                   terminate immediately                            /
     /                                                                    /
     //////////////////////////////////////////////////////////////////////

*****USER SAYS>  1

The lower and upper bounds for the RV ranges must be between
1000 and 10000

     The lower bound :
*****USER SAYS>  1000

     The upper bound :
*****USER SAYS>  10000

     //////////////////////////////////////////////////////////////////////
     /                                                                    /
     /  Let SIZE be an attack size in the RV range for which a robust     /
     /  solution is to be found.  The LP for the robust defense           /
     /  requires that the game values for SIZE be known.  Thus the        /
     /  set of attack sizes over which a robust defense is desired        /
     /  must be a subset of the attack sizes for the Basic Game.          /
     /                                                                    /
     //////////////////////////////////////////////////////////////////////

     //////////////////////////////////////////////////////////////////////
     /                                                                    /
     /  YROBUST is called to solve the robust defense.  Then XROBUST      /
     /  is called to find the optimal attacks and the expected            /
     /  survival rates.                                                   /
     /                                                                    /
     //////////////////////////////////////////////////////////////////////

XMAPS...WORDS OF MEMORY AVAILABLE
INTEGER:      100000          REAL:     50000

XMAPS...YOU HAVE ROOM FOR  27998 NON-ZEROS IN THE BASIS FACTORS
YOU COULD REDUCE REAL MEMORY FROM     100000 TO     89901

XMAPS...WORDS OF MEMORY AVAILABLE
INTEGER:        1000          REAL:      1000

XMAPS...YOU HAVE ROOM FOR   2987 NON-ZEROS IN THE BASIS FACTORS
YOU COULD REDUCE REAL MEMORY FROM       1000 TO        591

     //////////////////////////////////////////////////////////////////////
     /                                                                    /
     /                     More messages from XMP                         /
     /                                                                    /
     //////////////////////////////////////////////////////////////////////
```

```
        //////////////////////////////////////////////////////////////////
        /                                                                /
        /               SOLUTION REPORT FOR THE ROBUST GAME              /
        /                                                                /
        //////////////////////////////////////////////////////////////////

    THE ROBUST DEFENSE STRATEGY FOR RV RANGE  1000 TO 10000 :

   ****************************************************************************************************************

                                                 TARGET TYPE  1
             0         1         2         3         4         5         6         7         8         9
         ------------------------------------------------------------------------------------------------------
         : 0.2468 : 0.0066 : 0.1489 : 0.0402 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
         : 0.5575 :
         ------------------------------------------------------------------------------------------------------

   ****************************************************************************************************************
```

```
THE OPTIMAL ATTACK STRATEGIES AGAINST THE ROBUST DEFENSE

******************************************************************************************

               TARGET TYPE    1:  100.00% OF TOTAL TARGETS, WITH 100.00% OF TOTAL VALUE
ATTACK SIZE        0         1         2         3         4         5         6         7         8         9
------------------------------------------------------------------------------------------------------------------
    1000      : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 :
              ----------------------------------------------------------------------------------------------------
    2000      : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 :
              ----------------------------------------------------------------------------------------------------
    3000      : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 :
              ----------------------------------------------------------------------------------------------------
    4000      : 0.0000 : 0.0000 : 0.0000 : 0.6667 : 0.0000 : 0.0000 : 0.3333 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 :
              ----------------------------------------------------------------------------------------------------
    5000      : 0.0000 : 0.0000 : 0.0000 : 0.3333 : 0.0000 : 0.0000 : 0.6667 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 :
              ----------------------------------------------------------------------------------------------------
    6000      : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 :
              ----------------------------------------------------------------------------------------------------
    7000      : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 :
              : 0.0000 :
              ----------------------------------------------------------------------------------------------------
    8000      : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 :
              : 0.0000 :
              ----------------------------------------------------------------------------------------------------
    9000      : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 1.0000 :
              : 0.0000 :
              ----------------------------------------------------------------------------------------------------
   10000      : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 1.0000 :
------------------------------------------------------------------------------------------------------------------


******************************************************************************************
```

```
THE EXPECTED TARGET SURVIVAL RATE WITH THE ROBUST DEFENSE

******************************

 ATTACK SIZE
______________________________
     1000      :     0.8157
     2000      :     0.7021
     3000      :     0.5884
     4000      :     0.4822
     5000      :     0.3759
     6000      :     0.2697
     7000      :     0.1791
     8000      :     0.1191
     9000      :     0 0793
    10000      :     0.0528
______________________________

******************************
```

```
THE ROBUST DEFENSE ALLOCATION FOR RV RANGE  1000 TO 10000 :

*************************************************************************************************

                                     TARGET TYPE  1
          0        1        2        3        4        5        6        7        8        9
   ------------------------------------------------------------------------------------------
   :  246.8 :    6.6 :  148.9 :   40.2 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
   :  557.5 :
   ------------------------------------------------------------------------------------------

*************************************************************************************************
```

THE OPTIMAL ATTACK ALLOCATION AGAINST THE ROBUST DEFENSE

```
*************************************************************************************************

                                                   ATTACK SIZE =  1000
TARGET TYPE        0         1         2         3         4         5         6         7         8         9
---------------------------------------------------------------------------------------------------------------
     1       :    0.0 : 1000.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
             :    0.0 :
---------------------------------------------------------------------------------------------------------------

                                                   ATTACK SIZE =  2000
                   0         1         2         3         4         5         6         7         8         9
---------------------------------------------------------------------------------------------------------------
     1       :    0.0 :    0.0 : 1000.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
             :    0.0 :
---------------------------------------------------------------------------------------------------------------

                                                   ATTACK SIZE =  3000
                   0         1         2         3         4         5         6         7         8         9
---------------------------------------------------------------------------------------------------------------
     1       :    0.0 :    0.0 :    0.0 : 1000.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.6 :     0.0 :
             :    0.0 :
---------------------------------------------------------------------------------------------------------------

                                                   ATTACK SIZE =  4000
                   0         1         2         3         4         5         6         7         8         9
---------------------------------------------------------------------------------------------------------------
     1       :    0.0 :    0.0 :    0.0 :  666.7 :     0.0 :     0.0 :   333.3 :     0.0 :     0.0 :     0.0 :
             :    0.0 :
---------------------------------------------------------------------------------------------------------------

                                                   ATTACK SIZE =  5000
                   0         1         2         3         4         5         6         7         8         9
---------------------------------------------------------------------------------------------------------------
     1       :    0.0 :    0.0 :    0.0 :  333.3 :     0.0 :     0.0 :   666.7 :     0.0 :     0.0 :     0.0 :
             :    0.0 :
---------------------------------------------------------------------------------------------------------------

                                                   ATTACK SIZE =  6000
                   0         1         2         3         4         5         6         7         8         9
---------------------------------------------------------------------------------------------------------------
     1       :    0.0 :    0.0 :    0.0 :    0.0 :     0.0 :     0.0 :  1000.0 :     0.0 :     0.0 :     0.0 :
             :    0.0 :
---------------------------------------------------------------------------------------------------------------

                                                   ATTACK SIZE =  7000
                   0         1         2         3         4         5         6         7         8         9
---------------------------------------------------------------------------------------------------------------
     1       :    0.0 :    0.0 :    0.0 :    0.0 :     0.0 :     0.0 :     0.0 :  1000.0 :     0.0 :     0.0 :
             :    0.0 :
---------------------------------------------------------------------------------------------------------------
```

```
                                             ATTACK SIZE =  8000
                  0         1         2         3         4         5         6         7         8         9
    ----------------------------------------------------------------------------------------------------------------------
       1     :    0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 : 1000.0 :     0.0 :
             :    0.0 :
    ----------------------------------------------------------------------------------------------------------------------

                                             ATTACK SIZE =  9000
                  0         1         2         3         4         5         6         7         8         9
    ----------------------------------------------------------------------------------------------------------------------
       1     :    0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 : 1000.0 :
             :    0.0 :
    ----------------------------------------------------------------------------------------------------------------------

                                             ATTACK SIZE = 10000
                  0         1         2         3         4         5         6         7         8         9
    ----------------------------------------------------------------------------------------------------------------------
       1     :    0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
             : 1000.0 :
    ----------------------------------------------------------------------------------------------------------------------


    **********************************************************************************************************************
```

```
THE EXPECTED NUMBER OF TARGETS (SURVIVING) WITH THE ROBUST DEFENSE

•••••••••••••••••••••••••

                TARGET TYPE
 ATTACK SIZE  :      1
--------------------------
      1000    :   815.70
      2000    :   702.06
      3000    :   588.41
      4000    :   482.17
      5000    :   375.93
      6000    :   269.68
      7000    :   179.08
      8000    :   119.12
      9000    :    79.29
     10000    :    52.80
--------------------------

•••••••••••••••••••••••••

FORTRAN STOP

$
     ///////////////////////////////////////////////////////////////////////
     /                                                                     /
     /          TEST1.OUT contains a complete solution report              /
     /                                                                     /
     ///////////////////////////////////////////////////////////////////////
```

B. EXAMPLE 2

```
$ LINK BATCH
$ RUN BATCH

     /////////////////////////////////////////////////////////////////
     /                                                               /
     /      BATCH will be used to generate an input file for use     /
     /                          with RPPDM                           /
     /                                                               /
     /////////////////////////////////////////////////////////////////

Please type in the desired file name (of less than 10 characters
  including the extension) for storage of the parameters ?
*****USER SAYS>  TEST2.IN


You have three options for the output of the results :

     1) TERMINAL only
     2) FILE only
     3) TERMINAL and FILE

Please enter the number for the desired option ?
*****USER SAYS>  2

Please type in the desired file name (of less than 10 characters
  including the extension) ?
*****USER SAYS>  TEST2.OUT

The MAXIMUM number of RV's (up to 30) at a single target ?
*****USER SAYS>  15

The MAXIMUM number of INTERCEPTORS (up to 30) at a single target ?
*****USER SAYS>  19

Select one of the following attack methodologies:

     1) SIMULTANEOUS ATTACK
     2) SEQUENTIAL ATTACK

Please input the number of the desired attack ?
*****USER SAYS>  2

The FAILURE rate of the RV's ?
*****USER SAYS>  .2

Select one of the following defense methodologies:

     1) ONE SHOT
     2) SHOOT LOOK SHOOT

Please input the number for the desired option ?
*****USER SAYS>  2

Is the defender aware, afterthe attack begins, of the number
of RV's slated for each target (Y or N) ?
*****USER SAYS>  N

The FAILURE rate for the first salvo interceptors ?
```

```
*****USER SAYS>  .2

The FAILURE rate for the second salvo interceptors ?
*****USER SAYS>  .3

The MINIMUM attack size ?
*****USER SAYS>  1000

The MAXIMUM attack size ?
*****USER SAYS>  8000

The attack size INCREMENT ?
*****USER SAYS>  1000

The NUMBER of interceptors ?
*****USER SAYS>  4000

The TOTAL NUMBER of targets ?
*****USER SAYS>  1000

The number of TYPES of targets ?
*****USER SAYS>  3

Enter first the RELATIVE VALUE and then the NUMBER of targets
for each type. Separate the two entries for each target type with
a comma and hit <CR> following the entries for each target type:
*****USER SAYS>  1, 300
*****USER SAYS>  2, 400
*****USER SAYS>  3.5, 300

   ////////////////////////////////////////////////////////////////////
   /                                                                  /
   /  Both BATCH and RPPDM will check to make sure that the sum of    /
   /  numbers of targets for the target types equal the total num-    /
   /  be- cntered earlier.  The programs will terminate if the        /
   /  values are inconsistent.                                        /
   /                                                                  /
   ////////////////////////////////////////////////////////////////////
```

```
Please enter the number of different ranges of RV's for which robust
  solutions are to be found ?
         ***** Enter 0 if no robust solution is desired*****
*****USER SAYS>  2

The lower and upper bounds for the RV ranges must be between
1000 and  8000

     The lower bound :
*****USER SAYS>  1000

     The upper bound :
*****USER SAYS>  8000


NEXT


     The lower bound :
*****USER SAYS>  4000

     The upper bound :
*****USER SAYS>  8000

FORTRAN STOP
```

```
$ TYPE TEST2.IN
     ////////////////////////////////////////////////////////////////////
     /                                                                  /
     /                      PRINT OUT THE INPUT FILE                    /
     /                                                                  /
     ////////////////////////////////////////////////////////////////////
           2
 TEST2.OUT
          15
          19
           2
  0.20000000
           2
N
  0.20000000
  0.30000000
        1000
        8000
        1000
        4000
        1000
           3
  1.00000000      ,         300
  2.00000000      .         400
  3.50000000      .         300
           2
        1000
        8000
        4000
        8000
```

```
$ RUN RPPDM

THERE ARE TWO INPUT OPTIONS:

     1) TERMINAL (INTERACTIVE)
     2) FILE     (BATCH)

PLEASE ENTER THE NUMBER OF THE DESIRED OPTION ?
*****USER SAYS>  2

PLEASE ENTER THE NAME OF THE INPUT FILE (OF LESS THAN 12
  CHARACTERS INCLUDING THE EXTENSION) ?
*****USER SAYS>  TEST2.IN

XMAPS...WORDS OF MEMORY AVAILABLE
INTEGER:      60000            REAL:     60000

XMAPS...YOU HAVE ROOM FOR  27998 NON-ZEROS IN THE BASIS FACTORS
YOU COULD REDUCE REAL MEMORY FROM     60000 TO     43291

XMAPS...WORDS OF MEMORY AVAILABLE
INTEGER:     100000            REAL:     50000

XMAPS...YOU HAVE ROOM FOR  26798 NON-ZEROS IN THE BASIS FACTORS
YOU COULD REDUCE REAL MEMORY FROM    100000 TO     95151

XMAPS...WORDS OF MEMORY AVAILABLE
INTEGER:       1000            REAL:      1000

XMAPS...YOU HAVE ROOM FOR   2987 NON-ZEROS IN THE BASIS FACTORS
YOU COULD REDUCE REAL MEMORY FROM      1000 TO       661

XMAPS...WORDS OF MEMORY AVAILABLE
INTEGER:     100000            REAL:     50000

XMAPS...YOU HAVE ROOM FOR  26798 NON-ZEROS IN THE BASIS FACTORS
YOU COULD REDUCE REAL MEMORY FROM    100000 TO     95151

XMAPS...WORDS OF MEMORY AVAILABLE
INTEGER:       1000            REAL:      1000

XMAPS...YOU HAVE ROOM FOR   2987 NON-ZEROS IN THE BASIS FACTORS
YOU COULD REDUCE REAL MEMORY FROM      1000 TO       661


FORTRAN STOP
```

```
$ TYPE TEST2.OUT

     /////////////////////////////////////////////////////////////////////
     /                                                                   /
     /         The file TEST2.OUT is displayed on the terminal           /
     /                                                                   /
     /////////////////////////////////////////////////////////////////////

THE PARAMETERS OF THIS PREALLOCATED PREFERENTIAL DEFENSE GAME

**************************************************************************

 THE ATTACK METHODOLOGY                            SEQUENTIAL
                                                     WITH ATTACK SIZE AT
                                                     A TARGET UNKNOWN TO
                                                     THE DEFENDER
 THE DEFENSE METHODOLOGY                           SHOOT LOOK SHOOT

 THE FAILURE RATE OF RV'S                          0.200
 THE FAILURE RATE OF THE FIRST SALVO
      INTERCEPTORS                                 0.200
 THE FAILURE RATE OF THE SECOND SALVO
      INTERCEPTORS                                 0.300

 MAXIMUM NUMBER OF RV'S ATTACKING A
      SINGLE TARGET                                   15
 MAXIMUM NUMBER OF INTERCPTORS DEFENDING A
      SINGLE TARGET                                   19

 THE MINIMUM NUMBER OF RV'S                         1000
 THE MAXIMUM NUMBER OF RV'S                         8000
 THE ATTACK SIZE INCREMENT                          1000
 THE NUMBER OF INTERCEPTORS                         4000
 THE TOTAL NUMBER OF TARGETS                        1000

 THE NUMBER OF TARGET TYPES                            3
       TARGET TYPE  1:
           NUMBER OF TARGETS                         300
           RELATIVE VALUE                          1.000
       TARGET TYPE  2:
           NUMBER OF TARGETS                         400
           RELATIVE VALUE                          2.000
       TARGET TYPE  3:
           NUMBER OF TARGETS                         300
           RELATIVE VALUE                          3.500

**************************************************************************
```

THE ATTACKER'S BASIC GAME MINIMAX STRATEGIES

```
**************************************************************************************************************

                  TARGET TYPE    1:   30.00% OF TOTAL TARGETS, WITH  13.95% OF TOTAL VALUE
ATTACK SIZE         0         1         2         3         4         5         6         7         8         9
--------------------------------------------------------------------------------------------------------------
   1000        : 0.7619 : 0.0700 : 0.0717 : 0.0964 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0006 :
               -----------------------------------------------------------------------------------------------
   2000        : 0.5237 : 0.1401 : 0.1435 : 0.1927 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------------------
   3000        : 0.2856 : 0.2101 : 0.2152 : 0.2891 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------------------
   4000        : 0.0844 : 0.2693 : 0.2758 : 0.3705 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------------------
   5000        : 0.0000 : 0.2174 : 0.2226 : 0.2365 : 0.3100 : 0.0136 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------------------
   6000        : 0.0000 : 0.0000 : 0.3964 : 0.0893 : 0.2531 : 0.2569 : 0.0043 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------------------
   7000        : 0.0000 : 0.0000 : 0.3268 : 0.1037 : 0.0000 : 0.4887 : 0.0789 : 0.0019 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------------------
   8000        : 0.0000 : 0.0000 : 0.2705 : 0.0610 : 0.1920 : 0.0073 : 0.3340 : 0.1353 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
--------------------------------------------------------------------------------------------------------------


                  TARGET TYPE    2:   40.00% OF TOTAL TARGETS, WITH  37.21% OF TOTAL VALUE
                    0         1         2         3         4         5         6         7         8         9
--------------------------------------------------------------------------------------------------------------
   1000        : 0.7530 : 0.0350 : 0.0359 : 0.0381 : 0.0390 : 0.0466 : 0.0524 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------------------
   2000        : 0.5060 : 0.0700 : 0.0717 : 0.0762 : 0.0780 : 0.0931 : 0.1049 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------------------
   3000        : 0.2591 : 0.1051 : 0.1076 : 0.1143 : 0.1169 : 0.1397 : 0.1573 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------------------
   4000        : 0.0000 : 0.1347 : 0.1379 : 0.1681 : 0.0000 : 0.4075 : 0.0969 : 0.0549 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------------------
   5000        : 0.0000 : 0.0000 : 0.2459 : 0.0579 : 0.1653 : 0.0000 : 0.2293 : 0.2965 : 0.0051 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------------------
   6000        : 0.0000 : 0.0000 : 0.1982 : 0.0447 : 0.1265 : 0.1034 : 0.1110 : 0.0354 : 0.1887 : 0.1920 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------------------
   7000        : 0.0000 : 0.0000 : 0.1634 : 0.0368 : 0.1043 : 0.0853 : 0.0826 : 0.0882 : 0.0974 : 0.0533 :
               : 0.1415 : 0.1472 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------------------
   8000        : 0.0000 : 0.0000 : 0.1353 : 0.0305 : 0.0864 : 0.0802 : 0.0016 : 0.1676 : 0.0872 : 0.0000 :
          : 0.0000 : 0.1230 : 0.2883 : 0.0000 : 0.0000 : 0.0000 :
--------------------------------------------------------------------------------------------------

            TARGET TYPE    3:   30.00% OF TOTAL TARGETS, WITH  48.84% OF TOTAL VALUE
              0         1         2         3         4         5         6         7         8         9
--------------------------------------------------------------------------------------------------
 1000     : 0.7406 : 0.0200 : 0.0205 : 0.0218 : 0.0223 : 0.0266 : 0.0241 : 0.0249 : 0.0116 : 0.0384 :
          : 0.0493 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
          ----------------------------------------------------------------------------------------
 2000     : 0.4812 : 0.0400 : 0.0410 : 0.0435 : 0.0445 : 0.0532 : 0.0483 : 0.0497 : 0.0231 : 0.0768 :
          : 0.0985 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
          ----------------------------------------------------------------------------------------
 3000     : 0.2218 : 0.0600 : 0.0615 : 0.0653 : 0.0668 : 0.0798 : 0.0724 : 0.0746 : 0.0347 : 0.1152 :
          : 0.1478 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
          ----------------------------------------------------------------------------------------
 4000     : 0.0000 : 0.0769 : 0.0788 : 0.0837 : 0.0856 : 0.1023 : 0.0928 : 0.0953 : 0.0466 : 0.1444 :
          : 0.1882 : 0.0054 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
          ----------------------------------------------------------------------------------------
 5000     : 0.0000 : 0.0000 : 0.1405 : 0.0317 : 0.0897 : 0.0831 : 0.0035 : 0.1718 : 0.0898 : 0.0000 :
          : 0.0000 : 0.2214 : 0.1569 : 0.0117 : 0.0000 : 0.0000 :
          ----------------------------------------------------------------------------------------
 6000     : 0.0000 : 0.0000 : 0.1133 : 0.0255 : 0.0723 : 0.0591 : 0.0573 : 0.0649 : 0.0395 : 0.0856 :
          : 0.1052 : 0.0206 : 0.0000 : 0.0593 : 0.2740 : 0.0235 :
          ----------------------------------------------------------------------------------------
 7000     : 0.0000 : 0.0000 : 0.0000 : 0.1531 : 0.0000 : 0.0837 : 0.0290 : 0.0481 : 0.0391 : 0.0846 :
          : 0.0750 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.4874 :
          ----------------------------------------------------------------------------------------
 8000     : 0.0000 : 0.0000 : 0.0000 : 0.1290 : 0.0000 : 0.0485 : 0.0240 : 0.0927 : 0.0319 : 0.0000 :
          : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.6739 :
--------------------------------------------------------------------------------------------------


**************************************************************************************************
```

```
THE DEFENDER'S BASIC GAME MINIMAX STRATEGIES

*********************************************************************************************

                 TARGET TYPE    1:   30.00% OF TOTAL TARGETS, WITH  13.95% OF TOTAL VALUE
ATTACK SIZE         0        1        2        3        4        5        6        7        8        9
---------------------------------------------------------------------------------------------------------
    1000      : 0.3016 : 0.2058 : 0.4300 : 0.0626 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              -------------------------------------------------------------------------------------------
    2000      : 0.3016 : 0.2058 : 0.4300 : 0.0626 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              -------------------------------------------------------------------------------------------
    3000      : 0.3016 : 0.2058 : 0.4300 : 0.0626 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              -------------------------------------------------------------------------------------------
    4000      : 0.2611 : 0.2105 : 0.2758 : 0.2527 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              -------------------------------------------------------------------------------------------
    5000      : 0.2446 : 0.1010 : 0.2001 : 0.0196 : 0.4347 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              -------------------------------------------------------------------------------------------
    6000      : 0.4638 : 0.0000 : 0.0845 : 0.0785 : 0.0161 : 0.3572 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              -------------------------------------------------------------------------------------------
    7000      : 0.4952 : 0.0000 : 0.0624 : 0.0753 : 0.0000 : 0.0151 : 0.3521 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              -------------------------------------------------------------------------------------------
    8000      : 0.5939 : 0.0000 : 0.0381 : 0.0660 : 0.0070 : 0.0000 : 0.2495 : 0.0456 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
---------------------------------------------------------------------------------------------------------

                 TARGET TYPE    2:   40.00% OF TOTAL TARGETS, WITH  37.21% OF TOTAL VALUE
                    0        1        2        3        4        5        6        7        8        9
---------------------------------------------------------------------------------------------------------
    1000      : 0.1408 : 0.1111 : 0.1071 : 0.1140 : 0.0077 : 0.2549 : 0.2643 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              -------------------------------------------------------------------------------------------
    2000      : 0.1408 : 0.1111 : 0.1071 : 0.1140 : 0.0077 : 0.2549 : 0.2643 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              -------------------------------------------------------------------------------------------
    3000      : 0.1408 : 0.1111 : 0.1071 : 0.1140 : 0.0077 : 0.2549 : 0.2643 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              -------------------------------------------------------------------------------------------
    4000      : 0.1621 : 0.0966 : 0.0991 : 0.1093 : 0.0000 : 0.0219 : 0.5110 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              -------------------------------------------------------------------------------------------
    5000      : 0.4048 : 0.0000 : 0.0504 : 0.1246 : 0.0000 : 0.0000 : 0.0000 : 0.3665 : 0.0225 : 0.0312 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              -------------------------------------------------------------------------------------------
    6000      : 0.5184 : 0.0000 : 0.0272 : 0.0392 : 0.0316 : 0.0613 : 0.0608 : 0.0000 : 0.0000 : 0.0705 :
              : 0.1912 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              -------------------------------------------------------------------------------------------
    7000      : 0.5426 : 0.0000 : 0.0120 : 0.0359 : 0.0225 : 0.0138 : 0.0593 : 0.0675 : 0.0030 : 0.0000 :
              : 0.0000 : 0.0472 : 0.1961 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              -------------------------------------------------------------------------------------------
    8000      : 0.5584 : 0.0000 : 0.0032 : 0.0194 : 0.0308 : 0.0392 : 0.0000 : 0.0000 : 0.0660 : 0.0642 :
```

```
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.1260 : 0.0928 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
-----------------------------------------------------------------------------------------------------------

                     TARGET TYPE    3:   30.00% OF TOTAL TARGETS, WITH  48.84% OF TOTAL VALUE
                       0        1        2        3        4        5        6        7        8        9
-----------------------------------------------------------------------------------------------------------
         1000     : 0.0718 : 0.0595 : 0.0611 : 0.0652 : 0.0412 : 0.0310 : 0.1114 : 0.1144 : 0.0000 : 0.0000 :
                  : 0.0199 : 0.1244 : 0.3002 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                 ------------------------------------------------------------------------------------------
         2000     : 0.0718 : 0.0595 : 0.0611 : 0.0652 : 0.0412 : 0.0310 : 0.1114 : 0.1144 : 0.0000 : 0.0000 :
                  : 0.0199 : 0.1244 : 0.3002 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                 ------------------------------------------------------------------------------------------
         3000     : 0.0718 : 0.0595 : 0.0611 : 0.0652 : 0.0412 : 0.0310 : 0.1114 : 0.1144 : 0.0000 : 0.0000 :
                  : 0.0199 : 0.1244 : 0.3002 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                 ------------------------------------------------------------------------------------------
         4000     : 0.1813 : 0.0257 : 0.0546 : 0.0598 : 0.0364 : 0.0226 : 0.1013 : 0.1124 : 0.0000 : 0.0000 :
                  : 0.0053 : 0.0779 : 0.3227 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                 ------------------------------------------------------------------------------------------
         5000     : 0.3756 : 0.0000 : 0.0263 : 0.0314 : 0.0443 : 0.0265 : 0.0000 : 0.0581 : 0.0781 : 0.0442 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.2510 : 0.0645 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                 ------------------------------------------------------------------------------------------
         6000     : 0.5088 : 0.0000 : 0.0018 : 0.0237 : 0.0162 : 0.0132 : 0.0358 : 0.0375 : 0.0000 : 0.0000 :
                  : 0.0370 : 0.0546 : 0.0410 : 0.0000 : 0.0000 : 0.0000 : 0.0101 : 0.0000 : 0.0995 : 0.1207 :
                 ------------------------------------------------------------------------------------------
         7000     : 0.6058 : 0.0000 : 0.0000 : 0.0000 : 0.0273 : 0.0114 : 0.0267 : 0.0231 : 0.0000 : 0.0000 :
                  : 0.0347 : 0.0678 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.2031 :
                 ------------------------------------------------------------------------------------------
         8000     : 0.6656 : 0.0000 : 0.0000 : 0.0000 : 0.0275 : 0.0000 : 0.0000 : 0.0395 : 0.0178 : 0.0250 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.2246 :
-----------------------------------------------------------------------------------------------------------


***********************************************************************************************************
```

```
THE GAME VALUES

*****************************

 ATTACK SIZE
______________________________
     1000     :     0.8615
     2000     :     0.7229
     3000     :     0.5844
     4000     :     0.4473
     5000     :     0.3406
     6000     :     0.2707
     7000     :     0.2218
     8000     :     0.1836
______________________________

*****************************
```

THE ATTACKER'S BASIC GAME TARGET ALLOCATION

```
*****************************************************************************************

                                        ATTACK SIZE =  1000
TARGET TYPE      0        1        2        3        4        5        6        7        8        9
-----------------------------------------------------------------------------------------------------
     1      :  228.6 :   21.0 :   21.5 :   28.9 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
            :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
            -----------------------------------------------------------------------------------------
     2      :  301.2 :   14.0 :   14.3 :   15.2 :   15.6 :   18.6 :   21.0 :    0.0 :    0.0 :    0.0 :
            :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
            -----------------------------------------------------------------------------------------
     3      :  222.2 :    6.0 :    6.1 :    6.5 :    6.7 :    8.0 :    7.2 :    7.5 :    3.5 :   11.5 :
            :   14.8 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
-----------------------------------------------------------------------------------------------------

                                        ATTACK SIZE =  2000
                 0        1        2        3        4        5        6        7        8        9
-----------------------------------------------------------------------------------------------------
     1      :  157.1 :   42.0 :   43.0 :   57.8 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
            :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
            -----------------------------------------------------------------------------------------
     2      :  202.4 :   28.0 :   28.7 :   30.5 :   31.2 :   37.3 :   41.9 :    0.0 :    0.0 :    0.0 :
            :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
            -----------------------------------------------------------------------------------------
     3      :  144.4 :   12.0 :   12.3 :   13.1 :   13.4 :   16.0 :   14.5 :   14.9 :    6.9 :   23.0 :
            :   29.6 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
-----------------------------------------------------------------------------------------------------

                                        ATTACK SIZE =  3000
                 0        1        2        3        4        5        6        7        8        9
-----------------------------------------------------------------------------------------------------
     1      :   85.7 :   63.0 :   64.6 :   86.7 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
            :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
            -----------------------------------------------------------------------------------------
     2      :  103.6 :   42.0 :   43.0 :   45.7 :   46.8 :   55.9 :   62.9 :    0.0 :    0.0 :    0.0 :
            :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
            -----------------------------------------------------------------------------------------
     3      :   66.5 :   18.0 :   18.4 :   19.6 :   20.0 :   24.0 :   21.7 :   22.4 :   10.4 :   34.6 :
            :   44.3 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
-----------------------------------------------------------------------------------------------------

                                        ATTACK SIZE =  4000
                 0        1        2        3        4        5        6        7        8        9
-----------------------------------------------------------------------------------------------------
     1      :   25.3 :   80.8 :   82.7 :  111.1 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
            :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
            -----------------------------------------------------------------------------------------
     2      :    0.0 :   53.9 :   55.2 :   67.2 :    0.0 :  163.0 :   38.8 :   22.0 :    0.0 :    0.0 :
            :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
            -----------------------------------------------------------------------------------------
     3      :    0.0 :   23.1 :   23.6 :   25.1 :   25.7 :   30.7 :   27.8 :   28.6 :   14.0 :   43.3 :
            :   56.5 :    1.6 :    0.0 :    0.0 :    0.0 :    0.0 :
-----------------------------------------------------------------------------------------------------
```

```
                                      ATTACK SIZE = 5000
              0        1        2        3        4        5       6        7        8        9
-----------------------------------------------------------------------------------------------------
  1     :    0.0 :   65.2 :   66.8 :   70.9 :   93.0 :    4.1 :    0.0 :    0.0 :    0.0 :    0.0 :
        :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
        ---------------------------------------------------------------------------------------------
  2     :    0.0 :    0.0 :   98.4 :   23.1 :   66.1 :    0.0 :   91.7 :  118.6 :    2.1 :    0.0 :
        :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
        ---------------------------------------------------------------------------------------------
  3     :    0.0 :    0.0 :   42.2 :    9.5 :   26.9 :   24.9 :    1.1 :   51.5 :   26.9 :    0.0 :
        :    0.0 :   66.4 :   47.1 :    3.5 :    0.0 :    0.0 :
-----------------------------------------------------------------------------------------------------

                                      ATTACK SIZE = 6000
              0        1        2        3        4        5       6        7        8        9
-----------------------------------------------------------------------------------------------------
  1     :    0.0 :    0.0 :  118.9 :   26.8 :   75.9 :   77.1 :    1.3 :    0.0 :    0.0 :    0.0 :
        :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
        ---------------------------------------------------------------------------------------------
  2     :    0.0 :    0.0 :   79.3 :   17.9 :   50.6 :   41.4 :   44.4 :   14.2 :   75.5 :   76.8 :
        :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
        ---------------------------------------------------------------------------------------------
  3     :    0.0 :    0.0 :   34.0 :    7.7 :   21.7 :   17.7 :   17.2 :   19.5 :   11.9 :   25.7 :
        :   31.5 :    6.2 :    0.0 :   17.8 :   82.2 :    7.0 :
-----------------------------------------------------------------------------------------------------

                                      ATTACK SIZE = 7000
              0        1        2        3        4        5       6        7        8        9
-----------------------------------------------------------------------------------------------------
  1     :    0.0 :    0.0 :   98.0 :   31.1 :    0.0 :  146.6 :   23.7 :    0.6 :    0.0 :    0.0 :
        :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
        ---------------------------------------------------------------------------------------------
  2     :    0.0 :    0.0 :   65.4 :   14.7 :   41.7 :   34.1 :   33.1 :   35.3 :   38.9 :   21.3 :
        :   56.6 :   58.9 :    0.0 :    0.0 :    0.0 :    0.0 :
        ---------------------------------------------------------------------------------------------
  3     :    0.0 :    0.0 :    0.0 :   45.9 :    0.0 :   25.1 :    8.7 :   14.4 :   11.7 :   25.4 :
        :   22.5 :    0.0 :    0.0 :    0.0 :    0.0 :  146.2 :
-----------------------------------------------------------------------------------------------------

                                      ATTACK SIZE = 8000
              0        1        2        3        4        5       6        7        8        9
-----------------------------------------------------------------------------------------------------
  1     :    0.0 :    0.0 :   81.2 :   18.3 :   57.6 :    2.2 :  100.2 :   40.6 :    0.0 :    0.0 :
        :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
        ---------------------------------------------------------------------------------------------
  2     :    0.0 :    0.0 :   54.1 :   12.2 :   34.5 :   32.1 :    0.6 :   67.0 :   34.9 :    0.0 :
        :    0.0 :   49.2 :  115.3 :    0.0 :    0.0 :    0.0 :
        ---------------------------------------------------------------------------------------------
  3     :    0.0 :    0.0 :    0.0 :   38.7 :    0.0 :   14.5 :    7.2 :   27.8 :    9.6 :    0.0 :
        :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :  202.2 :
-----------------------------------------------------------------------------------------------------

*****************************************************************************************************
```

THE DEFENDER'S BASIC GAME TARGET ALLOCATION

ATTACK SIZE = 1000

TARGET TYPE	0	1	2	3	4	5	6	7	8	9
1	90.5	61.8	129.0	18.8	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	56.3	44.5	42.8	45.6	3.1	102.0	105.7	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	21.5	17.9	18.3	19.6	12.3	9.3	33.4	34.3	0.0	0.0
	6.0	37.3	90.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0

ATTACK SIZE = 2000

	0	1	2	3	4	5	6	7	8	9
1	90.5	61.8	129.0	18.8	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	56.3	44.5	42.8	45.6	3.1	102.0	105.7	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	21.5	17.9	18.3	19.6	12.3	9.3	33.4	34.3	0.0	0.0
	6.0	37.3	90.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0

ATTACK SIZE = 3000

	0	1	2	3	4	5	6	7	8	9
1	90.5	61.8	129.0	18.8	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	56.3	44.5	42.8	45.6	3.1	102.0	105.7	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	21.5	17.9	18.3	19.6	12.3	9.3	33.4	34.3	0.0	0.0
	6.0	37.3	90.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0

ATTACK SIZE = 4000

	0	1	2	3	4	5	6	7	8	9
1	78.3	63.1	82.7	75.8	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	64.9	38.7	39.6	43.7	0.0	8.7	204.4	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	54.4	7.7	16.4	17.9	10.9	6.8	30.4	33.7	0.0	0.0
	1.6	23.4	96.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0

ATTACK SIZE = 5000

	0	1	2	3	4	5	6	7	8	9
1	73.4 0.0	30.3 0.0	60.0 0.0	5.9 0.0	130.4 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0
2	161.9 0.0	0.0 0.0	20.2 0.0	49.8 0.0	0.0 0.0	0.0 0.0	0.0 0.0	146.6 0.0	9.0 0.0	12.5 0.0
3	112.7 0.0	0.0 0.0	7.9 0.0	9.4 0.0	13.3 75.3	7.9 19.3	0.0 0.0	17.4 0.0	23.4 0.0	13.3 0.0

ATTACK SIZE = 6000

	0	1	2	3	4	5	6	7	8	9
1	139.1 0.0	0.0 0.0	25.4 0.0	23.5 0.0	4.8 0.0	107.2 0.0	0.0 0.0	0.0 0.0	0.0 0.0	0.0 0.0
2	207.3 76.5	0.0 0.0	10.9 0.0	15.7 0.0	12.6 0.0	24.5 0.0	24.3 0.0	0.0 0.0	0.0 0.0	28.2 0.0
3	152.6 11.1	0.0 16.4	0.5 12.3	7.1 0.0	4.9 0.0	4.0 0.0	10.7 3.0	11.3 0.0	0.0 29.8	0.0 36.2

ATTACK SIZE = 7000

	0	1	2	3	4	5	6	7	8	9
1	148.6 0.0	0.0 0.0	18.7 0.0	22.6 0.0	0.0 0.0	4.5 0.0	105.6 0.0	0.0 0.0	0.0 0.0	0.0 0.0
2	217.1 0.0	0.0 18.9	4.8 78.4	14.4 0.0	9.0 0.0	5.5 0.0	23.7 0.0	27.0 0.0	1.2 0.0	0.0 0.0
3	181.7 10.4	0.0 20.3	0.0 0.0	0.0 0.0	8.2 0.0	3.4 0.0	8.0 0.0	6.9 0.0	0.0 0.0	0.0 60.9

ATTACK SIZE = 8000

	0	1	2	3	4	5	6	7	8	9
1	178.2 0.0	0.0 0.0	11.4 0.0	19.8 0.0	2.1 0.0	0.0 0.0	74.8 0.0	13.7 0.0	0.0 0.0	0.0 0.0
2	223.4 0.0	0.0 0.0	1.3 0.0	7.8 0.0	12.3 50.4	15.7 37.1	0.0 0.0	0.0 0.0	26.4 0.0	25.7 0.0
3	199.7 0.0	0.0 0.0	0.0 0.0	0.0 0.0	8.2 0.0	0.0 0.0	0.0 0.0	11.8 0.0	5.3 0.0	7.5 67.4

```
THE EXPECTED NUMBER OF TARGETS SURVIVING

******************************************

                        TARGET TYPE
 ATTACK SIZE  :      1         2         3
------------------------------------------
    1000      :   255.09    344.93    259.20
    2000      :   210.18    289.87    218.41
    3000      :   165.28    234.80    177.61
    4000      :   144.70    178.50    131.44
    5000      :   137.28    126.95     97.45
    6000      :   100.65    102.71     78.84
    7000      :    92.90     96.33     54.68
    8000      :    64.73     91.88     41.80
------------------------------------------

******************************************
```

```
THE ATTACKER'S BASIC GAME RV ALLOCATION BY TARGET TYPE

*********************************************

                          TARGET TYPE
 ATTACK SIZE  :       1        2        3
-----------------------------------------------
     1000     :      151      370      479
     2000     :      302      740      959
     3000     :      452     1109     1438
     4000     :      580     1567     1853
     5000     :      804     1928     2268
     6000     :     1015     2282     2703
     7000     :     1168     2675     3157
     8000     :     1344     3120     3536
-----------------------------------------------

*********************************************
```

```
THE DEFENDER'S BASIC GAME INTERCEPTOR ALLOCATION BY TARGET TYPE

*********************************************

                          TARGET TYPE
 ATTACK SIZE  :       1        2        3
-----------------------------------------------
     1000     :      376     1423     2201
     2000     :      376     1423     2201
     3000     :      376     1423     2201
     4000     :      456     1519     2025
     5000     :      690     1400     1910
     6000     :      676     1406     1918
     7000     :      762     1606     1632
     8000     :      635     1858     1506
-----------------------------------------------

*********************************************
```

```
THE ROBUST DEFENSE STRATEGY FOR RV RANGE  1000 TO  8000 :

************************************************************************************************************

                                            TARGET TYPE  1
           0        1        2        3        4        5        6        7        8        9
       -------------------------------------------------------------------------------------------
       : 0.3130 : 0.1817 : 0.1818 : 0.0257 : 0.0128 : 0.2850 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
       : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
       -------------------------------------------------------------------------------------------

                                            TARGET TYPE  2
           0        1        2        3        4        5        6        7        8        9
       -------------------------------------------------------------------------------------------
       : 0.3777 : 0.0333 : 0.0631 : 0.1802 : 0.0000 : 0.0451 : 0.0602 : 0.0000 : 0.0000 : 0.0416 :
       : 0.0461 : 0.1527 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
       -------------------------------------------------------------------------------------------

                                            TARGET TYPE  3
           0        1        2        3        4        5        6        7        8        9
       -------------------------------------------------------------------------------------------
       : 0.3852 : 0.0000 : 0.0407 : 0.0527 : 0.0068 : 0.0242 : 0.2018 : 0.0000 : 0.0000 : 0.0000 :
       : 0.0260 : 0.0414 : 0.0343 : 0.0057 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0776 : 0.1037 :
       -------------------------------------------------------------------------------------------

************************************************************************************************************
```

THE OPTIMAL ATTACK STRATEGIES AGAINST THE ROBUST DEFENSE

```
*****************************************************************************************************

                     TARGET TYPE    1:   30.00% OF TOTAL TARGETS, WITH  13.95% OF TOTAL VALUE
ATTACK SIZE             0         1         2         3         4         5         6         7         8         9
-----------------------------------------------------------------------------------------------------------------------
    1000          : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
    2000          : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
    3000          : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
    4000          : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
    5000          : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
    6000          : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
    7000          : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
    8000          : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
-----------------------------------------------------------------------------------------------------------------------

                     TARGET TYPE    2:   40.00% OF TOTAL TARGETS, WITH  37.21% OF TOTAL VALUE
                        0         1         2         3         4         5         6         7         8         9
-----------------------------------------------------------------------------------------------------------------------
    1000          : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
    2000          : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
    3000          : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
    4000          : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
    5000          : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
    6000          : 0.0000 : 0.1250 : 0.0000 : 0.8750 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
    7000          : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
    8000          : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
          TARGET TYPE    3:   30.00% OF TOTAL TARGETS, WITH  48.84% OF TOTAL VALUE
                0         1         2         3         4         5         6         7         8         9
 ------------------------------------------------------------------------------------------------------------
     1000     : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              ----------------------------------------------------------------------------------------------
     2000     : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              ----------------------------------------------------------------------------------------------
     3000     : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              ----------------------------------------------------------------------------------------------
     4000     : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              ----------------------------------------------------------------------------------------------
     5000     : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              ----------------------------------------------------------------------------------------------
     6000     : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              ----------------------------------------------------------------------------------------------
     7000     : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
              ----------------------------------------------------------------------------------------------
     8000     : 0.0000 : 0.0000 : 0.6667 : 0.0000 : 0.0000 : 0.0000 : 0.3333 : 0.0000 : 0.0000 : 0.0000 :
              : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
 ------------------------------------------------------------------------------------------------------------

 ************************************************************************************************************
```

```
THE EXPECTED TARGET SURVIVAL RATE WITH THE ROBUST DEFENSE

*****************************

 ATTACK SIZE
-----------------------------
     1000      :      0.6828
     2000      :      0.5661
     3000      :      0.4577
     4000      :      0.3503
     5000      :      0.2752
     6000      :      0.2307
     7000      :      0.1872
     8000      :      0.1438
-----------------------------

*****************************
```

```
THE ROBUST DEFENSE ALLOCATION FOR RV RANGE  1000 TO  8000 :

••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••

                                        TARGET TYPE  1
          0         1         2         3         4         5         6         7         8         9
      -------------------------------------------------------------------------------------------------
      :  93.9 :   54.5 :   54.5 :    7.7 :    3.8 :   85.5 :    0.0 :    0.0 :    0.0 :    0.0 :
      :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
      -------------------------------------------------------------------------------------------------

                                        TARGET TYPE  2
          0         1         2         3         4         5         6         7         8         9
      -------------------------------------------------------------------------------------------------
      : 151.1 :   13.3 :   25.3 :   72.1 :    0.0 :   18.0 :   24.1 :    0.0 :    0.0 :   16.6 :
      :  18.4 :   61.1 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
      -------------------------------------------------------------------------------------------------

                                        TARGET TYPE  3
          0         1         2         3         4         5         6         7         8         9
      -------------------------------------------------------------------------------------------------
      : 115.6 :    0.0 :   12.2 :   15.8 :    2.0 :    7.3 :   60.5 :    0.0 :    0.0 :    0.0 :
      :   7.8 :   12.4 :   10.3 :    1.7 :    0.0 :    0.0 :    0.0 :    0.0 :   23.3 :   31.1 :
      -------------------------------------------------------------------------------------------------

••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
```

```
THE OPTIMAL ATTACK ALLOCATION AGAINST THE ROBUST DEFENSE

**************************************************************************************

                                           ATTACK SIZE =  1000
TARGET TYPE       0        1        2        3        4        5        6        7        8        9
     1       :   0.0 :  300.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
             :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
     2       :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
             :   0.0 :    0.0 :    0.0 :  400.0 :    0.0 :    0.0 :
     3       :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
             :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :

                                           ATTACK SIZE =  2000
                  0        1        2        3        4        5        6        7        8        9
     1       :   0.0 :  300.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
             :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
     2       :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
             :   0.0 :    0.0 :    0.0 :    0.0 :  400.0 :    0.0 :
     3       :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
             :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :

                                           ATTACK SIZE =  3000
                  0        1        2        3        4        5        6        7        8        9
     1       :   0.0 :    0.0 :  300.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
             :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
     2       :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
             :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
     3       :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
             :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :

                                           ATTACK SIZE =  4000
                  0        1        2        3        4        5        6        7        8        9
     1       :   0.0 :    0.0 :  300.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
             :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
     2       : 400.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
             :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
     3       :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
             :   0.0 :    0.0 :    0.0 :    0.0 :    0.0 :    0.0 :
```

```
                                         ATTACK SIZE =  5000
                  0         1         2         3         4         5         6         7         8         9
-----------------------------------------------------------------------------------------------------------------
    1      :     0.0 :     0.0 :     0.0 :   300.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           ------------------------------------------------------------------------------------------------------
    2      :     0.0 :   400.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           ------------------------------------------------------------------------------------------------------
    3      :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
-----------------------------------------------------------------------------------------------------------------

                                         ATTACK SIZE =  6000
                  0         1         2         3         4         5         6         7         8         9
-----------------------------------------------------------------------------------------------------------------
    1      :     0.0 :     0.0 :     0.0 :   300.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           ------------------------------------------------------------------------------------------------------
    2      :     0.0 :    50.0 :     0.0 :   350.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           ------------------------------------------------------------------------------------------------------
    3      :   300.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
-----------------------------------------------------------------------------------------------------------------

                                         ATTACK SIZE =  7000
                  0         1         2         3         4         5         6         7         8         9
-----------------------------------------------------------------------------------------------------------------
    1      :     0.0 :     0.0 :     0.0 :     0.0 :   300.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           ------------------------------------------------------------------------------------------------------
    2      :     0.0 :     0.0 :     0.0 :   400.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           ------------------------------------------------------------------------------------------------------
    3      :     0.0 :     0.0 :   300.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
-----------------------------------------------------------------------------------------------------------------

                                         ATTACK SIZE =  8000
                  0         1         2         3         4         5         6         7         8         9
-----------------------------------------------------------------------------------------------------------------
    1      :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :   300.0 :     0.0 :     0.0 :     0.0 :
           :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           ------------------------------------------------------------------------------------------------------
    2      :     0.0 :     0.0 :     0.0 :   400.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           ------------------------------------------------------------------------------------------------------
    3      :     0.0 :     0.0 :   200.0 :     0.0 :     0.0 :     0.0 :   100.0 :     0.0 :     0.0 :     0.0 :
           :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
-----------------------------------------------------------------------------------------------------------------

*****************************************************************************************************************
```

```
THE EXPECTED NUMBER OF TARGETS (SURVIVING) WITH THE ROBUST DEFENSE

*********************************************

                          TARGET TYPE
 ATTACK SIZE  :      1          2          3
---------------------------------------------
      1000    :   211.75       0.30       0.00
      2000    :   211.75       0.06       0.00
      3000    :   141.79       0.00       0.00
      4000    :   141.79     400.00       0.00
      5000    :    91.06     270.36       0.00
      6000    :    91.06     188.73     300.00
      7000    :    63.04     177.06     179.43
      8000    :     7.00     177.06     152.87
---------------------------------------------

*********************************************
```

```
THE DEFENDER'S ROBUST GAME INTERCEPTOR ALLOCATION BY TARGET TYPE

*********************************************

                          TARGET TYPE
 ATTACK SIZE  :      1         2         3
---------------------------------------------
     N/A      :     630       1521      1850
---------------------------------------------

*********************************************


THE ATTACKER'S ROBUST GAME RV ALLOCATION BY TARGET TYPE

*********************************************

                          TARGET TYPE
 ATTACK SIZE  :      1         2         3
---------------------------------------------
    1000      :     300       5200         0
    2000      :     300       5600         0
    3000      :     600          0         0
    4000      :     600          0         0
    5000      :     900        400         0
    6000      :     900       1100         0
    7000      :    1200       1200       600
    8000      :    1800       1200      1000
---------------------------------------------

*********************************************
```

```
THE ROBUST DEFENSE STRATEGY FOR RV RANGE  4000 TO  8000 :

****************************************************************************************************

                                           TARGET TYPE  1
        0         1         2         3         4         5         6         7         8         9
    ------------------------------------------------------------------------------------------------
    : 0.5145 : 0.0000 : 0.1156 : 0.0389 : 0.1918 : 0.1391 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
    : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
    ------------------------------------------------------------------------------------------------

                                           TARGET TYPE  2
        0         1         2         3         4         5         6         7         8         9
    ------------------------------------------------------------------------------------------------
    : 0.4972 : 0.0000 : 0.0306 : 0.0407 : 0.0328 : 0.0637 : 0.0632 : 0.0000 : 0.0000 : 0.0732 :
    : 0.1986 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
    ------------------------------------------------------------------------------------------------

                                           TARGET TYPE  3
        0         1         2         3         4         5         6         7         8         9
    ------------------------------------------------------------------------------------------------
    : 0.4873 : 0.0000 : 0.0042 : 0.0247 : 0.0168 : 0.0137 : 0.0372 : 0.0390 : 0.0000 : 0.0000 :
    : 0.0384 : 0.0567 : 0.0426 : 0.0000 : 0.0000 : 0.0000 : 0.0105 : 0.0000 : 0.1033 : 0.1254 :
    ------------------------------------------------------------------------------------------------

****************************************************************************************************
```

THE OPTIMAL ATTACK STRATEGIES AGAINST THE ROBUST DEFENSE

```
**************************************************************************************************

                   TARGET TYPE    1:   30.00% OF TOTAL TARGETS, WITH  13.95% OF TOTAL VALUE
ATTACK SIZE          0        1        2        3        4        5        6        7        8        9
--------------------------------------------------------------------------------------------------
    1000       : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------
    2000       : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------
    3000       : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------
    4000       : 0.0000 : 0.0000 : 0.3333 : 0.0000 : 0.0000 : 0.6667 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------
    5000       : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------
    6000       : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------
    7000       : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------
    8000       : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
--------------------------------------------------------------------------------------------------


                   TARGET TYPE    2:   40.00% OF TOTAL TARGETS, WITH  37.21% OF TOTAL VALUE
                     0        1        2        3        4        5        6        7        8        9
--------------------------------------------------------------------------------------------------
    1000       : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------
    2000       : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 :
               -----------------------------------------------------------------------------------
    3000       : 0.5000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------
    4000       : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------
    5000       : 0.4167 : 1.0000 : 0.0000 : 0.5833 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------
    6000       : 0.0000 : 0.1250 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------
    7000       : 0.0000 : 0.0000 : 0.0000 : 1.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
               -----------------------------------------------------------------------------------
    8000       : 0.0000 : 0.0000 : 0.0000 : 0.5000 : 0.0000 : 0.5000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
```

```
                  TARGET TYPE    3:   30.00% OF TOTAL TARGETS, WITH  48.84% OF TOTAL VALUE
                       0         1         2         3         4         5         6         7         8         9
-----------------------------------------------------------------------------------------------------------------------
      1000        : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
      2000        : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
      3000        : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
      4000        : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
      5000        : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
      6000        : 1.0000 : 0.0000 : 0.0000 : 0.2381 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
      7000        : 0.0000 : 0.0000 : 1.0000 : 0.7143 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
                  -----------------------------------------------------------------------------------------------------
      8000        : 0.0000 : 0.0000 : 0.6667 : 1.0000 : 0.0000 : 0.0000 : 0.3333 : 0.0000 : 0.0000 : 0.0000 :
                  : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 : 0.0000 :
-----------------------------------------------------------------------------------------------------------------------


*****************************************************************************************************************
```

```
THE EXPECTED TARGET SURVIVAL RATE WITH THE ROBUST DEFENSE

*****************************

 ATTACK SIZE
______________________________
      1000      :      0.5944
      2000      :      0.4969
      3000      :      0.4404
      4000      :      0.3838
      5000      :      0.3272
      6000      :      0.2707
      7000      :      0.2141
      8000      :      0.1575
______________________________

*****************************
```

```
THE ROBUST DEFENSE ALLOCATION FOR RV RANGE  4000 TO  8000 :

*****************************************************************************************

                                  TARGET TYPE  1
         0         1         2         3         4         5         6         7         8         9
   ------------------------------------------------------------------------------------------------------
   :  154.3 :    0.0 :    34.7 :    11.7 :    57.6 :    41.7 :     0.0 :     0.0 :     0.0 :     0.0 :
   :    0.0 :    0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
   ------------------------------------------------------------------------------------------------------

                                  TARGET TYPE  2
         0         1         2         3         4         5         6         7         8         9
   ------------------------------------------------------------------------------------------------------
   :  198.9 :    0.0 :    12.2 :    16.3 :    13.1 :    25.5 :    25.3 :     0.0 :     0.0 :    29.3 :
   :   79.4 :    0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
   ------------------------------------------------------------------------------------------------------

                                  TARGET TYPE  3
         0         1         2         3         4         5         6         7         8         9
   ------------------------------------------------------------------------------------------------------
   :  146.2 :    0.0 :     1.3 :     7.4 :     5.1 :     4.1 :    11.2 :    11.7 :     0.0 :     0.0 :
   :   11.5 :   17.0 :    12.8 :     0.0 :     0.0 :     0.0 :     3.2 :     0.0 :    31.0 :    37.6 :
   ------------------------------------------------------------------------------------------------------

*****************************************************************************************
```

```
THE OPTIMAL ATTACK ALLOCATION AGAINST THE ROBUST DEFENSE

*****************************************************************************************************

                                            ATTACK SIZE =  1000
TARGET TYPE        0         1         2         3         4         5       6         7         8         9
-----------------------------------------------------------------------------------------------------------
    1        :   0.0 :   300.0 :     0.0 :     0.0 :     0.0 :     0.0 :   0.0 :     0.0 :     0.0 :     0.0 :
             :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           ------------------------------------------------------------------------------------------------
    2        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :   0.0 :     0.0 :     0.0 :     0.0 :
             :   0.0 :     0.0 :     0.0 :   400.0 :     0.0 :     0.0 :
           ------------------------------------------------------------------------------------------------
    3        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :   0.0 :     0.0 :     0.0 :     0.0 :
             :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
-----------------------------------------------------------------------------------------------------------

                                            ATTACK SIZE =  2000
                   0         1         2         3         4         5       6         7         8         9
-----------------------------------------------------------------------------------------------------------
    1        :   0.0 :     0.0 :   300.0 :     0.0 :     0.0 :     0.0 :   0.0 :     0.0 :     0.0 :     0.0 :
             :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           ------------------------------------------------------------------------------------------------
    2        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :   0.0 :     0.0 :     0.0 :     0.0 :
             :   0.0 :     0.0 :     0.0 :     0.0 :   400.0 :     0.0 :
           ------------------------------------------------------------------------------------------------
    3        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :   0.0 :     0.0 :     0.0 :     0.0 :
             :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
-----------------------------------------------------------------------------------------------------------

                                            ATTACK SIZE =  3000
                   0         1         2         3         4         5       6         7         8         9
-----------------------------------------------------------------------------------------------------------
    1        :   0.0 :     0.0 :   300.0 :     0.0 :     0.0 :     0.0 :   0.0 :     0.0 :     0.0 :     0.0 :
             :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           ------------------------------------------------------------------------------------------------
    2        : 200.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :   0.0 :     0.0 :     0.0 :     0.0 :
             :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           ------------------------------------------------------------------------------------------------
    3        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :   0.0 :     0.0 :     0.0 :     0.0 :
             :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
-----------------------------------------------------------------------------------------------------------

                                            ATTACK SIZE =  4000
                   0         1         2         3         4         5       6         7         8         9
-----------------------------------------------------------------------------------------------------------
    1        :   0.0 :     0.0 :   100.0 :     0.0 :     0.0 :   200.0 :   0.0 :     0.0 :     0.0 :     0.0 :
             :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           ------------------------------------------------------------------------------------------------
    2        : 400.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :   0.0 :     0.0 :     0.0 :     0.0 :
             :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
           ------------------------------------------------------------------------------------------------
    3        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :   0.0 :     0.0 :     0.0 :     0.0 :
             :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
-----------------------------------------------------------------------------------------------------------
```

```
                                     ATTACK SIZE =  5000
            0         1         2         3         4         5         6         7         8         9
---------------------------------------------------------------------------------------------------------
  1     :   0.0 :     0.0 :     0.0 :   300.0 :     0.0 :   300.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        -------------------------------------------------------------------------------------------------
  2     : 166.7 :   400.0 :     0.0 :   233.3 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        -------------------------------------------------------------------------------------------------
  3     :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
---------------------------------------------------------------------------------------------------------

                                     ATTACK SIZE =  6000
            0         1         2         3         4         5         6         7         8         9
---------------------------------------------------------------------------------------------------------
  1     :   0.0 :     0.0 :     0.0 :   300.0 :     0.0 :   300.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        -------------------------------------------------------------------------------------------------
  2     :   0.0 :    50.0 :     0.0 :   400.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        -------------------------------------------------------------------------------------------------
  3     : 300.0 :     0.0 :     0.0 :    71.4 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
---------------------------------------------------------------------------------------------------------

                                     ATTACK SIZE =  7000
            0         1         2         3         4         5         6         7         8         9
---------------------------------------------------------------------------------------------------------
  1     :   0.0 :     0.0 :     0.0 :     0.0 :   300.0 :   300.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        -------------------------------------------------------------------------------------------------
  2     :   0.0 :     0.0 :     0.0 :   400.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        -------------------------------------------------------------------------------------------------
  3     :   0.0 :     0.0 :   300.0 :   214.3 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
---------------------------------------------------------------------------------------------------------

                                     ATTACK SIZE =  8000
            0         1         2         3         4         5         6         7         8         9
---------------------------------------------------------------------------------------------------------
  1     :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :   300.0 :   300.0 :     0.0 :     0.0 :     0.0 :
        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        -------------------------------------------------------------------------------------------------
  2     :   0.0 :     0.0 :     0.0 :   200.0 :     0.0 :   200.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
        -------------------------------------------------------------------------------------------------
  3     :   0.0 :     0.0 :   200.0 :   300.0 :     0.0 :     0.0 :   100.0 :     0.0 :     0.0 :     0.0 :
        :   0.0 :     0.0 :     0.0 :     0.0 :     0.0 :     0.0 :
---------------------------------------------------------------------------------------------------------

*********************************************************************************************************
```

```
THE EXPECTED NUMBER OF TARGETS (SURVIVING) WITH THE ROBUST DEFENSE

*********************************************

                         TARGET TYPE
ATTACK SIZE  :      1          2          3
---------------------------------------------
    1000     :   170.93       0.10       0.00
    2000     :   132.24       0.02       0.00
    3000     :   132.24     200.00       0.00
    4000     :    59.28     400.00       0.00
    5000     :   118.56     503.91       0.00
    6000     :   118.56     202.10     334.42
    7000     :    82.08     172.52     258.24
    8000     :    27.36     148.20     285.64
---------------------------------------------

*********************************************
```

```
THE DEFENDER'S ROBUST GAME INTERCEPTOR ALLOCATION BY TARGET TYPE

*********************************************

                          TARGET TYPE
 ATTACK SIZE  :      1         2         3
---------------------------------------------
    N/A       :     543      1463.     1994
---------------------------------------------

*********************************************


THE ATTACKER'S ROBUST GAME RV ALLOCATION BY TARGET TYPE

*********************************************

                          TARGET TYPE
 ATTACK SIZE  :      1         2         3
---------------------------------------------
    1000      :     300      5200         0
    2000      :     600      5600         0
    3000      :     600         0         0
    4000      :    1200         0         0
    5000      :    2400      1100         0
    6000      :    2400      1250       214
    7000      :    2700      1200      1243
    8000      :    3300      1600      1900
---------------------------------------------

*********************************************

$
    /////////////////////////////////////////////////////////////////
    /                                                               /
    /  If desired, the user may now edit TEST2.IN to change some of /
    /           the parameters and rerun RPPDM                      /
    /                                                               /
    /////////////////////////////////////////////////////////////////
```

REFERENCES

1. Bracken, Jerome, Brooks, Peter S. and Falk, James E., Robust Preallocated Preferential Defense, P-1816, Institute for Defense Analyses, August 1985.

2. Bracken, Jerome, Falk, James E and Tai, A.J. Allen, Robustness of Preallocated Preferential Defense With Assumed Attack Size and Perfect Attacking and Defending Weapons, P-1860, Institute for Defense Analyses, September 1986.

3. Hogg, Christopher J., OPUS1 Reference Manual, Teledyne-Brown Engineering, August 1981.

4. Key, John C., MVADEM User's Guide and Reference Manual: Revision 1, URH2S-01, Sparta, Inc., April 1984.

5. Matheson, John D., Preferential Strategies, AR 66-2, Analytic Services, Inc., May 1966.

6. Matheson, John D., Preferential Strategies with Imperfect Information, AR 67-1, Analytic Services, Inc., April 1967.

7. Matheson, John D., Multidimensional Preferential Strategies, SDN 75-3, Analytic Services, Inc., November 1975.

APPENDIX A

PROGRAM MAIN

```
                  PROGRAM MAIN
c
c       MAIN will solve for the basic game and the Case II,II as
c       outlined in Bracken, Brooks, and Falk [1].  Eith batch or
c       interactive modes may be employed. XMP is the linear programming
c       package used.
c
c       Parameter statements for establishing limits on the problem's
c       size
c
         PARAMETER (MAXNAT=25, MAXR=30, MAXS=30, MAXNTYPE=7)
c
c       *****************************************************************
c         MAXNAT     THE MAXIMUM ALLOWABLE NUMBER OF ATTACK SIZES
c          MAXR      THE MAXIMUM ALLOWABLE 'R'
c          MAXS      THE MAXIMUM ALLOWABLE 'S'
c        MAXNTYPE    THE MAXIMUM ALLOWABLE TYPES OF TARGETS
c       *****************************************************************
c
c       Variable type declarations for the basic game
c
         REAL XBG(MAXNAT, MAXNTYPE, 0:MAXR)
         REAL YBG(MAXNAT, MAXNTYPE, 0:MAXS), VBG(MAXNAT)
         REAL VTYPE(MAXNTYPE), VFRAC(MAXNTYPE), NFRAC(MAXNTYPE)
         REAL P(0:MAXR, 0:MAXS), TS(MAXNAT, MAXNTYPE)
         INTEGER R, S,MAXRV,MINRV,INCRV,INT,TARGETS
         INTEGER RV(MAXNAT),A,NTAR(MAXNTYPE)
         INTEGER TMXBG(MAXNAT,MAXNTYPE),TMYBG(MAXNAT,MAXNTYPE)
c
c       *****************************************************************
c           XBG      THE ATTACKER'S OPTIMAL BASIC GAME STRATEGY
c           YBG      THE DEFENDER'S OPTIMAL BASIC GAME STRATEGY
c           VBG      THE GAME VALUE
c            TS      THE EXPECTED NUMBER OF TARGETS (SURVIVING)
c            P       THE PIJ MATRIX
c            R       THE MAXIMUM NUMBER OF RV'S AT A SINGLE TARGET
c            S       THE MAXIMUM NUMBER OF INTERCEPTORS AT A SINGLE
c                        TARGET
c          MAXRV     THE MAXIMUM ATTACK SIZE
c          MINRV     THE MINIMUM ATTACK SIZE
c          INCRV     THE ATTACK SIZE INCREMENT
c           INT      THE NUMBER OF INTERCEPTORS
c         TARGETS    THE NUMBER OF TARGETS
c            RV      THE ATTACK SIZES
c            A       THE NUMBER OF ATTACK SIZES
c          VTYPE     THE RELATIVE VALUES OF THE TARGET TYPES
c           NTAR     THE NUMBER OF TARGETS FOR EACH TYPE
c          VFRAC     THE FRACTION OF TOTAL VALUE FOR EACH TYPE
c          NFRAC     THE FRACTION OF TOTAL TARGETS FOR EACH TYPE
c           TS       TARGET EQUIVALENTS OF EXPECTED 'VBG' AND 'VII'
c          TMXBG     RV ALLOC. FOR BASIC GAME BY TARGET TYPES
c          TMYBG     INTERCEPTOR ALLOC. FOR BASIC GAME BY TARGET TYPES
c
c       *****************************************************************
```

```
c
c       Variable type declarations for Case II.II
c
          REAL YII(MAXNTYPE, 0:MAXS)
          REAL XII(MAXNAT,MAXNTYPE,0:MAXR), VII(MAXNAT)
          REAL VBGR(MAXNAT)
          INTEGER MAXRRV, MINRRV
          INTEGER RRV(MAXNAT), AR
          INTEGER TMXII(MAXNAT,MAXNTYPE), TMYII(MAXNAT,MAXNTYPE)
c
c       ****************************************************************
c          YII       THE ROBUST DEFENDER'S STRATEGY
c          XII       THE OPTIMAL ATTACKER'S STRATEGY AGAINST YII
c          VII       THE EXPECTED SURVIVAL RATE GIVEN YII AND XII
c          VBGR      THE GAME VALUE FOR ROBUST ATTACK SIZE RANGE
c         MAXRRV     THE MAXIMUM ATTACK SIZE IN THE ROBUST RANGE
c         MINRRV     THE MINIMUM ATTACK SIZE IN THE ROBUST RANGE
c          RRV       THE ATTACK SIZES IN THE ROBUST RANGE
c           AR       THE NUMBER OF ATTACK SIZES IN THE ROBUST RANGE
c         TMXII      RV ALLOC. FOR BASIC GAME BY TARGET TYPES
c         TMYII      INTERCEPTOR ALLOC. FOR BASIC GAME BY TARGET TYPES
c       ****************************************************************
c
c
          INTEGER TER,IN,FILE,OUT,OUT1,OUT2,QPR
          CHARACTER*1 ANSWER, NAME*80, FILEOUT*12, FILEIN*12
c
c       ****************************************************************
c          TER       THE I/O UNIT NUMBER FOR THE TERMINAL
c          IN        THE I/O UNIT NUMBER FOR THE INPUT DEVICE
c          FILE      THE I/O UNIT NUMBER FOR AN OUTPUT FILE
c          OUT       THE I/O UNIT NUMBER FOR THE OUTPUT DEVICE
c          QPR       THE I/O UNIT NUMBER FOR DISPLAY OF PROMPTS
c         ANSWER     'Y' OR 'N' ANSWER TO SELECTED QUESTIONS
c          NAME      HEADER FOR OUTPUT
c         FILEOUT    NAME OF AN OUTPUT FILE
c       ****************************************************************
c
c       Set up common statement for input, IOIN, error output, IOERR,
c       results output, IOLOG (XMP only)
c
          COMMON/IO_UNIT/IOIN, IOERR, IOLOG
          IOIN=5
          IOERR=5
          IOLOG=5
c
c       Set TER to default I/O unit number for terminal
c
          TER=5
c
c       Set FILE to TER+1
c
          FILE=TER+1
c
```

```
c       Set IN to the I/O unit number for the input file/device
c
          WRITE(TER,'(A)') 'OTHERE ARE TWO INPUT OPTIONS:'
          WRITE(TER,'(A)') '0     1) TERMINAL (INTERACTIVE)'
          WRITE(TER,'(A)') '      2) FILE     (BATCH)'
          WRITE(TER,'(A)') 'OPLEASE ENTER THE NUMBER OF THE DESIRED OPTI
     1ON ?'
          READ(TER,*) NIN
          IF (NIN .NE. 1) THEN
              WRITE(TER,'(A)') 'OPLEASE ENTER THE NAME OF THE INPUT FILE
     1(OF LESS THAN 12'
              WRITE(TER,'(A)') ' CHARACTERS INCLUDING THE EXTENSION) ?'
              READ(TER, '(A)') FILEIN
c
c           Open file for file input
c
              OPEN(UNIT-TER+2, FILE-FILEIN, STATUS-'OLD')
              IN-TER+2
c
c           Open junk file for unnecessary output
c
              OPEN(UNIT-TER+3, FILE-'JUNK.DAT', STATUS-'SCRATCH')
              QPR-TER+3
          ELSE
              QPR-TER
              IN-TER
          ENDIF
c
c       Select output option
c
          WRITE(QPR,'(A)') '1You have three options for the output of th
     1e results : '
          WRITE(QPR,'(A)') '0     1) TERMINAL only'
          WRITE(QPR,'(A)') '      2) FILE only'
          WRITE(QPR,'(A)') '      3) TERMINAL and FILE'
          WRITE(QPR,'(A)') 'OPlease enter the number for the desired opt
     1ion ?'
          READ(IN, FMT-*) NOUT
c
c       Ask for file name, if necessary
c
          IF (NOUT .NE. 1) THEN
              WRITE(QPR,'(A)') 'OPlease type in the desired file name (o
     1f less than 10 characters'
              WRITE(QPR,'(A)') '  including the extension) ?'
                READ(IN,'(A)') FILEOUT
              OPEN(UNIT-FILE, FILE-FILEOUT, STATUS-'NEW')
          ENDIF
c
c       Input R and S
c
          WRITE(UNIT-QPR, FMT-'(A,I2,A)') 'OThe MAXIMUM number of RV''s
     1(up to ',MAXR,') at a single target ? '
          READ(UNIT-IN, FMT-*) R
```

```
      WRITE(UNIT=QPR, FMT='(A,I2,A)') 'OThe MAXIMUM number of INTERC
   1EPTORS (up to ',MAXS,') at a single target ? '
      READ(UNIT=IN, FMT=*) S
C
C    Input failure rates for
C
C    Select attack methodology
C
      WRITE(UNIT=QPR, FMT='(A)') 'OSelect one of the following attac
   1k methodologies: '
      WRITE(UNIT=QPR, FMT='(A)') '0     1) SIMULTANEOUS ATTACK'
      WRITE(UNIT=QPR, FMT='(A)') '      2) SEQUENTIAL ATTACK'
      WRITE(UNIT=QPR, FMT='(A)') 'OPlease input the number of the de
   1sired attack ? '
      READ(UNIT=IN, FMT=*) NATTYPE
C
C    Input the failure rate for the RV''s
C
      WRITE(UNIT=QPR, FMT='(A)') 'OThe FAILURE rate of the RV''s ? '
      READ(UNIT=IN, FMT=*) PFA
C
C   Select defense methodology
C
      WRITE(UNIT=QPR, FMT='(A)') 'OSelect one of the following defen
   1se methodologies: '
      WRITE(UNIT=QPR, FMT='(A)') '0     1) ONE SHOT'
      WRITE(UNIT=QPR, FMT='(A)') '      2) SHOOT LOOK SHOOT'
      WRITE(UNIT=QPR, FMT='(A)') 'OPlease input the number for the d
   1esired option ? '
      READ(UNIT=IN, FMT=*) NDFTYPE
C
C   If the the attack is sequential, find out whether the defender knows,
C   after the attack begins, the number of RV's slated for each target.
C
      IF (NATTYPE .EQ. 2) THEN
        WRITE(UNIT=QPR, FMT='(A)') 'OIs the defender aware, after
   1the attack begins, of the number'
        WRITE(UNIT=QPR, FMT='(A)') ' of RV''s slated for each targ
   1et (Y or N) ?'
        READ(UNIT=IN, FMT='(A)') ANSWER
        IF (ANSWER .EQ. 'Y') THEN
        IF (NDFTYPE .EQ. 1) THEN
          WRITE(UNIT=QPR, FMT='(A)')'ONOTE: This scenario is equ
   1ivalent to one with a simultaneous attack.'
          NATTYPE=1
        ENDIF
        ENDIF
      ENDIF
C
C    Find the failure rates for the interceptors
C
      IF (NDFTYPE .EQ. 1) THEN
        WRITE(UNIT=QPR, FMT='(A)') 'OThe FAILURE rate of the inter
   1ceptors ? '
```

```
              READ(UNIT-IN, FMT-*) PFD
          ELSE
              WRITE(UNIT-QPR, FMT-'(A)') 'OThe FAILURE rate for the firs
     lt salvo interceptors ? '
              READ(UNIT-IN, FMT-*) PFD1
              WRITE(UNIT-QPR, FMT-'(A)') 'OThe FAILURE rate for the seco
     lnd salvo interceptors ? '
              READ(UNIT-IN, FMT-*) PFD2
           ENDIF
C
C       Generate the appropriate pij's
C
          IF (NATTYPE .EQ. 1 .AND. NDFTYPE .EQ. 1) THEN
C
C           Simultaneous attack without SLS
C
              CALL SIMAT1(MAXR, MAXS, PFA, PFD, R, S, P)
C
          ELSE IF (NATTYPE .EQ. 1 .AND. NDFTYPE .EQ. 2) THEN
C
C           Simultaneous attack with SLS
C
              CALL SIMAT2(MAXR, MAXS, PFA, PFD1, PFD2, R, S, P)
C
          ELSE IF (NATTYPE .EQ. 2 .AND. NDFTYPE .EQ. 1 .AND.
     1ANSWER .EQ. 'N') THEN
C
C           Sequential attack without SLS and attack at target unknown
C
              CALL SEQAT1(MAXR, MAXS, PFA, 1.0, PFD, R, S, P)

          ELSE IF (NATTYPE .EQ. 2 .AND. NDFTYPE .EQ. 2 .AND.
     1ANSWER .EQ. 'N') THEN
C
C           Sequential attack with SLS and attack at target unknown
C
              CALL SEQAT1(MAXR, MAXS, PFA, PFD1, PFD2, R, S, P)

          ELSE IF (NATTYPE .EQ. 2 .AND. NDFTYPE .EQ. 2 .AND.
     1ANSWER .EQ. 'Y') THEN
C
C           Sequential attack with SLS and attack at target known
C
              CALL SEQAT2(MAXR, MAXS, PFA, PFD1, PFD2, R, S, P)

          ENDIF
C
C       Specific parameters of the game
C
          WRITE(QPR,'(A)') 'OThe MINIMUM attack size ?'
          READ(IN, FMT-*) MINRV
          WRITE(QPR,'(A)') 'OThe MAXIMUM attack size ?'
          READ(IN, FMT-*) MAXRV
          WRITE(QPR,'(A)') 'OThe attack size INCREMENT ?'
```

```
           READ(IN, FMT=*) INCRV
           WRITE(QPR,'(A)') 'OThe NUMBER of interceptors ?'
           READ(IN, fmt=*) INT
           WRITE(QPR,'(A)') 'OThe TOTAL NUMBER of targets ?'
           READ(IN, FMT=*) TARGETS
           WRITE(QPR,'(A)') 'OThe number of TYPES of targets ?'
           READ(IN, FMT=*) NTYPE
c
c        Only one target type
c
           IF (NTYPE .EQ. 1) THEN
               VTYPE(1)=1
               NTAR(1)=TARGETS
               TVT=TARGETS
c
c        More than one target type
c
           ELSE
               WRITE(QPR, '(A)') 'OEnter first the RELATIVE VALUE and the
     1n the NUMBER of targets'
               WRITE(QPR, '(A)') ' for each type. Separate the two entrie
     1s for each target type with'
               WRITE(QPR, '(A)') ' a comma and hit <CR> following the ent
     1ries for each target type: '
c
c            Loop through each target type
c
               DO 100 I = 1, NTYPE
                   READ(IN, fmt=*) VTYPE(I), NTAR(I)
                   TVT=TVT+VTYPE(I)*NTAR(I)
                   TNT=TNT+NTAR(I)
100            CONTINUE
               IF (TNT .NE. TARGETS) THEN
                   WRITE(TER, '(A)') 'OThe sum of the targets in the indi
     1vidual target types does not'
                   WRITE(TER, '(A)') ' equal the total number of targets'
                   STOP
               ENDIF
           ENDIF
c
c        Calculate the fractional values for VTYPE and NTAR
c
           DO 110 I = 1, NTYPE
               VFRAC(I)=VTYPE(I)*NTAR(I)/TVT
               NFRAC(I)=REAL(NTAR(I))/TARGETS
110        CONTINUE
c
c        Place the attack sizes in an array
c
           A=1+(MAXRV-MINRV)/INCRV
           DO 120 I= 1, A
               RV(I)=MINRV+(I-1)*INCRV
120        CONTINUE
c
```

```
c       Call Subroutine BG to solve for the basic game
c
          CALL BG(XBG, YBG, VBG, R, S, P,
     1RV, A, INT, TARGETS,
     1MAXNAT, MAXR, MAXS, MAXNTYPE, NTYPE,
     1VFRAC, NFRAC)
c
c       Print out results of the basic game
c
          IF (NOUT .EQ. 1) THEN
              OUT1-TER
              OUT2-TER
          ELSE IF (NOUT .EQ. 2) THEN
              OUT1-FILE
              OUT2-FILE
          ELSE
              OUT1-TER
              OUT2-FILE
          ENDIF
c
c       Calculate some statistics for output
c
          DO 130 K-1,A
              DO 131 NT-1, NTYPE
                  TS(K,NT)-0
                  TMX-0
                  TMY-0
                  DO 132 I-0, R
                      DO 133 J-0, S
                          TS(K,NT)-XBG(K,NT,I)*P(I,J)*YBG(K,NT,J)+
     1TS(K,NT)
                          IF (I .EQ. 0) THEN
                              TMY-TMY+J*YBG(K,NT,J)
                          ENDIF
133                  CONTINUE
                      TMX-TMX+I*XBG(K,NT,I)
132              CONTINUE
                  TS(K,NT)-TS(K,NT)*NTAR(NT)
                  TMXBG(K,NT)-NINT(TMX*NTAR(NT))
                  TMYBG(K,NT)-NINT(TMY*NTAR(NT))
131           CONTINUE
130       CONTINUE
c
c
          DO 140 OUT-OUT1, OUT2
c
              CALL SUMMARY(NATTYPE,NDFTYPE,ANSWER,MINRV,MAXRV,INCRV,
     1                     INT,TARGETS,R,S,PFA,PFD,PFD1,PFD2,OUT,
     1                     MAXNTYPE,NTYPE,VTYPE,NTAR)
c
              NAME-'1THE ATTACKER''S BASIC GAME MINIMAX STRATEGIES  '
              CALL STPRINT(NAME,XBG,RV,A,R,OUT,MAXNAT,MAXR,MAXNTYPE,
     1                     NTYPE,VFRAC,NFRAC)
c
```

```
              NAME-'1THE DEFENDER''S BASIC GAME MINIMAX STRATEGIES   '
              CALL STPRINT(NAME,YBG,RV,A,S,OUT,MAXNAT,MAXS,MAXNTYPE,
     1                     NTYPE,VFRAC,NFRAC)
c
              NAME-'1THE GAME VALUES   '
              CALL VPRINT(NAME,VBG,RV,A,OUT,MAXNAT)
c
              NAME-'1THE ATTACKER''S BASIC GAME TARGET ALLOCATION   '
              CALL ALPRINT(NAME,XBG,RV,A,R,OUT,MAXNAT,MAXR,MAXNTYPE,
     1                     NTYPE,VFRAC,NFRAC,TARGETS)
c
              NAME-'1THE DEFENDER''S BASIC GAME TARGET ALLOCATION   '
              CALL ALPRINT(NAME,YBG,RV,A,S,OUT,MAXNAT,MAXS,MAXNTYPE,
     1                     NTYPE,VFRAC,NFRAC,TARGETS)
c
              NAME-'1THE EXPECTED NUMBER OF TARGETS SURVIVING   '
              CALL ALVPRINT(NAME,VBG,RV,A,OUT,MAXNAT,
     1                      MAXNTYPE,NTYPE,TS)
c
              IF (NTYPE .NE. 1) THEN
                  ROB-.FALSE.
                  NAME-'1THE ATTACKER''S BASIC GAME RV ALLOCATION BY TAR
     1GET TYPE'
                  CALL RVINTCOUNT(NAME, TMXBG, RV, A, OUT,
     1                            MAXNAT,MAXNTYPE,NTYPE,ROB)
c
                  WRITE(OUT,'(/)')
                  NAME-'OTHE DEFENDER''S BASIC GAME INTERCEPTOR ALLOCATI
     1ON BY TARGET TYPE'
                  CALL RVINTCOUNT(NAME, TMYBG, RV, A, OUT,
     1                            MAXNAT, MAXNTYPE, NTYPE, ROB)
              ENDIF

140       CONTINUE
c
c       Ask for the number of robust solutions desired
c
              WRITE(QPR,'(A)') '1Please enter the number of different ra
     1nges of RV''s for which robust '
              WRITE(QPR,'(A)') '    solutions are to be found ?'
              WRITE(QPR,'(A)') '          ***** Enter 0 if no robust solu
     1tion is desired*****         '
              READ(IN,*) J
          IF (J .EQ. 0) THEN
              STOP
          ENDIF
          WRITE(QPR,'(A)') 'OThe lower and upper bounds for the RV range
     1s must be between'
          WRITE(QPR,'(I5,'' and '',I5)') RV(1), RV(A)
c
          DO 200 I1-1, J
              WRITE (QPR,'(A)') 'O     The lower bound :'
              READ (IN,*) MINRRV
              WRITE (QPR,'(A)') 'O     The upper bound :'
              READ (IN,*) MAXRRV
              AR=1+(MAXRRV-MINRRV)/INCRV
              DO 210 I=1, AR
210               RRV(I)=MINRRV+(I-1)*INCRV
              DO 220 I=1, A
                  IF (MINRRV .EQ. RV(I)) THEN
                      DO 221 I2=1, AR
221                       VBGR(I2)=VBG(I+I2-1)
                  ENDIF
220           CONTINUE
c
c           Find the robust defender's strategy, YII
c
              CALL YROBUST(YII, VBGR, R, S, P,
     1RRV, AR, INT, TARGETS,
     1MAXNAT,MAXR,MAXS,MAXNTYPE,NTYPE,NFRAC,VFRAC)
c
c           Find the XII's and the VII's
c
              CALL XROBUST(XII,YII,VII,R,S,P,
     1RV,A,INT,TARGETS,
     1MAXNAT,MAXR,MAXS,MAXNTYPE,NTYPE,NFRAC,VFRAC)
c
              DO 230 K3=1,A
                  DO 231 NT3=1, NTYPE
                      TS(K3,NT3)=0
                      TMX=0
                      DO 232 I3=0, R
                          DO 233 J3=0, S
                              TS(K3,NT3)=XII(K3,NT3,I3)*P(I3,J3)*
     1YII(NT3,J3)+TS(K3,NT3)
233                       CONTINUE
                          TMX=TMX+I3*XII(K3,NT3,I3)
232                  CONTINUE
                      TS(K3,NT3)=TS(K3,NT3)*NTAR(NT3)
                      TMXII(K3,NT3)=NINT(TMX*NTAR(NT3))
231              CONTINUE
230          CONTINUE
              DO 240 NT3=1, NTYPE
                  TMY=0
                  DO 241 J3=0,S
                      TMY=TMY+J3*YII(NT3,J3)
241               CONTINUE
                  TMYII(1,NT3)=NINT(TMY*NTAR(NT3))

240           CONTINUE
c
c           Print out the robust solution
c
              DO 250 OUT=OUT1, OUT2, FILE=TER
c
                  CALL YRPRINT(YII, MINRRV, MAXRRV, MAXS, S, OUT,
     I                           MAXNTYPE, NTYPE)
c
```

```
                  NAME='1THE OPTIMAL ATTACK STRATEGIES AGAINST THE ROBUS
     1T DEFENSE'
                  CALL STPRINT(NAME,XII,RV,A,R,OUT,MAXNAT,MAXR,MAXNTYPE,
     1                         NTYPE,VFRAC,NFRAC)
C
                  NAME='1THE EXPECTED TARGET SURVIVAL RATE WITH THE ROBU
     1ST DEFENSE'
                  CALL VPRINT(NAME,VII,RV,A,OUT,MAXNAT)
C
                  CALL ALYRPRINT(YII, MINRRV, MAXRRV, MAXS, S, OUT,
     1                         MAXNTYPE, NTYPE,NFRAC, TARGETS)
C
                  NAME='1THE OPTIMAL ATTACK ALLOCATION AGAINST THE ROBUS
     1T DEFENSE'
                  CALL ALPRINT(NAME,XII,RV,A,R,OUT,MAXNAT,MAXR,MAXNTYPE
     1                         ,NTYPE,VFRAC,NFRAC,TARGETS)
C
                  NAME='1THE EXPECTED NUMBER OF TARGETS (SURVIVING) WITH
     1 THE ROBUST DEFENSE'
                  CALL ALVPRINT(NAME,VII,RV,A,OUT,MAXNAT,MAXNTYPE,NTYPE,
     1                          TS)
C
                  IF (NTYPE .NE. 1) THEN
                      ROB=.TRUE.
                      NAME='1THE DEFENDER''S ROBUST GAME INTERCEPTOR ALL
     1OCATION BY TARGET TYPE'
                      CALL RVINTCOUNT(NAME, TMYII, RV, A, OUT,
     1                                MAXNAT,MAXNTYPE,NTYPE,ROB)
                      WRITE(OUT, '(/)')
C
                      ROB=.FALSE.
                      NAME='0THE ATTACKER''S ROBUST GAME RV ALLOCATION B
     1Y TARGET TYPE'
                      CALL RVINTCOUNT(NAME, TMXII, RV, A, OUT,
     1                                MAXNAT,MAXNTYPE,NTYPE,ROB)
                  ENDIF
C
250           CONTINUE
200       CONTINUE

          STOP
          END
```

APPENDIX B

SUBROUTINES SIMAT1, SIMAT2, SEQAT1, AND SEQAT2

```
        SUBROUTINE SIMAT1(MAXR,MAXS,PFA,PFD,R,S,P)
C
C    Subroutine SIMAT1 solves for the probability of a target
C    surviving under simultaneous attack and one shot defense
C
C    INPUT variables:
C
C       MAXR          INTEGER              The maximum allowable 'R'
C       MAXS          INTEGER              The maximum allowable 'S'
C       PFA           REAL                 The probability that a
C                                            missile will fail to destroy
C                                            a target.
C       PFD           REAL                 The probability that an
C                                            interceptor will fail to
C                                            destroy a missile.
C        R            INTEGER              The maximum number of missiles
C                                            that can attack a target.
C        S            INTEGER              The maximum number of
C                                            interceptors that can attack
C                                            a target.
C    OUTPUT variable:
C
C        P(I,J)       REAL                 The probability that a target
C                                            attacked by I missiles and
C                                            protected by J interceptors
C                                            will survive.
C
C
     INTEGER R, S
     REAL P(0:MAXR,0:MAXS), MNFRA
C
     A=1-PFA
     D=1-PFD
C
C    Loop through number of attacking missiles
C
     DO 10 I = 0, R
C
C      Loop through number of interceptors
C
       DO 20 J = 0, S
C
C        Assign value to P(I,J)
C
         IF (I .EQ. 0) THEN
             P(I,J) = 1
         ELSE IF (A .EQ. 1.0 .AND. D .EQ. 1.0) THEN
             IF (I .GT. J) THEN
                 P(I, J) = 0
             ELSE
                 P(I, J) = 1
             ENDIF
         ELSE
             MNINT = INT(J/I)
```

```
                      MNFRA = REAL(J)/REAL(I)-MNINT
                      P1 = (1-A*(1.0-D)**(MNINT+1))**(I*MNFRA)
                      P2 = (1-A*(1.0-D)**MNINT)**(I*(1-MNFRA))
                      P(I,J) = P1*P2
                  ENDIF
c
20            CONTINUE
10        CONTINUE
c
          RETURN
          END
```

```
      SUBROUTINE SIMAT2(MAXR,MAXS,PFA,PFD1,PFD2,
     1                  R,S,P)
c
c     Subroutine SIMAT2 calculates for the probability of a target
c     surviving under simultaneous attack and shoot, look, shoot
c     defense.
c
c     INPUT variables:
c
c       MAXR               INTEGER        The maximum allowable 'R'
c       MAXS               INTEGER        The maximum allowable 'S'
c       PFA                REAL           The probability that a
c                                           missile will fail to
c                                           destroy a target.
c       PFD1               REAL           The probability that a first
c                                           salvo interceptor will fail
c                                           to destroy a missile
c       PFD2               REAL           The probability that a second
c                                           salvo interceptor will fail
c                                           to destroy a missile
c        R                 INTEGER        The maximum number of
c                                           missiles that can attack a
c                                           target
c        S                 INTEGER        The maximum number of
c                                           interceptors that can
c                                           defend a target
c
c     OUTPUT variables:
c
c       P(I,J)             REAL           The probability that a target
c                                           attacked by I missiles and
c                                           protected by J interceptors
c                                           will survive
c
c
      INTEGER R,S
      REAL P(0:MAXR,0:MAXS)
C
C     DEFINITION OF LOCAL VARIABLES:
C
C          INPUTS ARE (1) MAX NUMBER OF
C          ATTACKERS N, (2) MAX NO. OF INTERCEPTORS M, (3) MAX NO.
C          OF SALVOS SAL, (4) SINGLE-SHOT INTERCEPTOR KILL PROBABILITY
C          ON EACH SALVO PKILL(1),...,PKILL(SAL), AND (5) ATTACKER
C          RELIABILITY RHO.  OUTPUTS ARE (1) MINIMUM PROBABILITY OF
C          TARGET DESTRUCTION WHEN K SALVOS REMAIN, S2(I,J),
C          I=1,...,M; J=1,...,N; K=1,...,SAL, (2) OPTIMAL NUMBER
C          OF INTERCEPTORS TO USE UNDER EACH CONDITION D(I,J), AND
C          (3) EXPECTED NUMBER OF INTERCEPTORS REMAINING AT THE
C          END OF THE ATTACK W2(I,J).  IN CASE OF TIES, THE
C          CHOICE IS MADE TO MAXIMIZE W2(I,J).
C
C     ARRAYS NEEDED ARE S1(M+1,N+1),S2(M+1,N+1), D(M+1,N+1),
C     PA(M+1,N+1), RR(N+1,N), T0(N), T1(N), PKILL(R),
C         W1(M+1,N+1), AND W2(M+1,N+1).
C
            DIMENSION S1(51,51),S2(51,51),D(51,51),W1(51,51),W2(51,51),
     1      PA(51,51),RR(51,50),T0(50),T1(50),PKILL(5)
            INTEGER D,SAL
C
C     SET LOCAL INPUT VARIABLES TO THEIR CORRESPONDING 'SHARED' INPUT
C     VARIABLES
C
      N=R
      M=S
      SAL=2
      RHO=1.-PFA
      PKILL(1)=1.-PFD1
      PKILL(2)=1.-PFD2
C
C     PKILL(1) IS THE S-S KILL PROB. ON THE FIRST SALVO,...,
C     PKILL(2) IS THE S-S KILL PROB. ON THE SECOND SALVO.
C
C     INITIALIZE; DO CASE OF 1 SALVO LEFT.
C
      RHOBAR = 1. - RHO
      M1=M+1
      N1=N+1
      DO 200 I=1,M1
      W2(I,1) = I-1
  200 S2(I,1) = 0.
      S2(1,2)=RHO
      DO 210 J=3,N1
      J1=J-1
  210 S2(1,J)=1.-RHOBAR*(1.-S2(1,J1))
      DO 211 I=1,M1
      DO 211 J=2,N1
      W2(I,J) = 0.0
  211 CONTINUE
      Q=1.-PKILL(SAL)
      DO 220 J=2,N1
      IF(J .GE. 2) GO TO 1211
      DO 1200 I=2,M1
      D(I,J) = I-1
 1200 S2(I,2) = Q**(I-1)
      GO TO 220
 1211 J1=J-1
      ZJ1=J1
      DO 220 I=2,M1
      I1=I-1
      ZI1=I1
      ZIJ=ZI1/ZJ1
      INTIJ=I1/J1
      ZINTIJ=INTIJ+1.E-2
      IF (ZINTIJ .LT. ZIJ) GO TO 212
C
C     ZIJ IS AN INTEGER
C
```

```
      S2(I,J)=1.-(1.-RHO*(Q**INTIJ))**J1
      GO TO 214
  212 CONTINUE
C
C     ZIJ IS NOT AN INTEGER
C
      INTIJ1=INTIJ+1
      IEXP1=I1-J1*INTIJ
      IEXP2=J1-IEXP1
      F1=(1.-RHO*(Q**INTIJ1))**IEXP1
      F2=(1.-RHO*(Q**INTIJ))**IEXP2
      S2(I,J)=1.-F1*F2
  214 CONTINUE
      D(I,J)=I1
  220 CONTINUE
      K=1
C
C     NOW WRITE RESULTS FOR K=1
C
  230 CONTINUE
      IF (SAL .EQ. 1) GO TO 900
C
C     MAIN LOOP FOR K SALVOES, K>1
C
      DO 420 K=2,SAL
      KK = SAL-K+1
      Q=1.-PKILL(KK)
C
      DO 300 I=1,M1
      DO 300 J=1,N1
      W1(I,J) = W2(I,J)
  300 S1(I,J) = S2(I,J)
C
C     FIRST FIX J
C
      DO 400 J=2,N1
      J1=J-1
      ZJ1=J1
C
C     CALCULATE ALL PA(-,-) VALUES
C
      PA(1,J)=1.
      DO 310 JJ=1,J1
  310 PA(1,JJ)=0.
C
C     FIRST CALCULATE T0 AND T1 VECTORS
C
      DO 380 I=2,M1
      I1=I-1
      ZI1=I1
      ZIJ=ZI1/ZJ1
      INTIJ=I1/J1
      ZINTIJ=INTIJ+1.E-5
      IF (ZINTIJ .LT. ZIJ) GO TO 330
```

```
C
C     ZIJ IS AN INTEGER
C
      FAC=Q**INTIJ
      DO 320 L=1,J1
      T1(L)=FAC
  320 TO(L)=1.-FAC
      GO TO 335
  330 CONTINUE
C
C     ZIJ IS NOT AN INTEGER
C
      LIM=J1-I1+J1*INTIJ
      LIM1=LIM+1
      FAC=Q**INTIJ
      DO 332 L=1,LIM
      T1(L)=FAC
  332 TO(L)=1.-FAC
      FAC=Q*FAC
      DO 333 L=LIM1,J1
      T1(L)=FAC
  333 TO(L)=1.-FAC
  335 CONTINUE
C
C     NOW FINISHED WITH TO AND T1
C
      RR(1,1)=TO(1)
      RR(2,1)=T1(1)
      IF (J1 .EQ. 1) GO TO 360
      DO 340 L=2,J1
      L1=L+1
      LM1=L-1
      RR(1,L)=RR(1,LM1)*TO(L)
  340 RR(L1,L)=RR(L,LM1)*T1(L)
      DO 350 L=2,J1
      LM1=L-1
      DO 350 J2=2,L
      JM1=J2-1
      RR(J2,L)=RR(J2,LM1)*TO(L)+RR(JM1,LM1)*T1(L)
  350 CONTINUE
  360 CONTINUE
      DO 370 JJ=1,J
  370 PA(I,JJ)=RR(JJ,J1)
  380 CONTINUE
C
C     ALL PA(-,-) VALUES ARE NOW READY
C
C     TEST IS TENTATIVE VALUE OF S(I,J)
C     ITEST IS TENTATIVE VALUE OF D(I,J)
C     WTEST IS TENTATIVE VALUE OF W(I,J)
C
  382 CONTINUE
      DO 395 I=2,M1
      TEST=1.
      DO 390 II=1,I
      MI=I-II+1
      FAC = 0.0
      DO 385 JJ=1,J
  385 FAC=FAC+PA(II,JJ)*S1(MI,JJ)
      TEST=AMIN1(FAC,TEST)
C
C     UPDATE TENTATIVE VALUES
C
  390 CONTINUE
      S2(I,J)=TEST
  395 CONTINUE
  400 CONTINUE
C
C     NOW WRITE RESULTS
C
  420 CONTINUE
C
  500 CONTINUE
C
C     STORE PIJ'S
C
      DO 154 I=0,N
      DO 155 J=0,M
         P(I,J)=1-S2(J+1,I+1)
  155 CONTINUE
  154 CONTINUE

      RETURN
  900 END
```

```
              SUBROUTINE SEQAT1(MAXR,MAXS,PFA,PFD1,PFD2,R,S,P)
c
c       Subroutine SEQAT1 solves for the probability of a target
c       surviving under unknown sequential attack and one shot or shoot,
c       look, shoot defense
c
c       INPUT vadriables:
c
c         MAXR             INTEGER              The maximum allowable 'R'
c         MAXS             INTEGER              The maximum allowable 'S'
c         PFA              REAL                 The probability that a
c                                                 missile will fail to
c                                                 destroy a target
c         PFD1              REAL                The probability that a
c                                                 first salvo interceptor
c                                                 will fail to destroy a
c                                                 missile
c         PFD2              REAL                The probability that a
c                                                 second salvo interceptor
c                                                 will fail to destroy a
c                                                 missile
c          R                INTEGER             The maximum number of
c                                                 missiles that can attack
c                                                 a target
c          S                INTEGER             The maximum number of
c                                                 interceptors that may
c                                                 attack a target
c
c       OUTPUT variable:
c
c         P(I,J)           REAL                 The probability that a
c                                                 target attacked by I
c                                                 missiles and protected
c                                                 by J interceptors will
c                                                 survive
c
c
          INTEGER R,S
          REAL P(0:MAXR, 0:MAXS)
c
c       LOCAL VARIABLE DEFINITIONS
c
c            RR(.,D) IS THE EXPECTED DAMAGE CURVE WITH ONE SHOT LEFT
c            Q(.,D) IS THE EXPECTED DAMAGE CURVE WITH TWO SHOTS LEFT
c            SLOPE(D) YIELDS PRIM READ SLOPE
c            NQ(D) IS THE NUMBER OF INT TO SHOOT AT 1ST RV, 1ST VOLLEY
c            NR(D) IS THE NUMBER OF INT TO SHOOT AT NEXT RV, 2ND VOLLEY
c            NRR(D) IS THE NUMBER REMAINING IF MISS (D-NQ(D))
c
      DIMENSION Q(0:100,0:100),RR(0:100,0:100),SLOPE(0:100),
     1          NR(0:100),NQ(0:100),RTRY(0:100),QTRY(0:100),
     1          NRR(0:100)
c
      CAPESS=1.-PFA
```

```
c
c   SET Q(A,0)=RR(A,0)=1-PFA**A
c
         DO IA=0,R
           IF(PFA.LT.0.000001) THEN
             IF(IA.EQ.0) Q(IA,0)=0.
             IF(IA.GT.0) Q(IA,0)=1.
           ELSE
             Q(IA,0)=1.-(PFA**IA)
           END IF
           RR(IA,0)=Q(IA,0)
         END DO
         NQ(0)=0
         NR(0)=0
c
c   SET SLOPE(D)=1.-PFA (UNPROTECTED VALUE)
c   AND SET RR(0,D)=Q(0,D)=0
c
         DO ID=0,S
           SLOPE(ID)=CAPESS
           Q(0,ID)=0.
           RR(0,ID)=0
         END DO
c
c   GENERATE RR(.,D) AND Q(.,D) FOR D=1,S
c
         DO ID=1,S
           SLPMAX=0.
           PROD=PFD2*CAPESS
           DO IA=1,R
             RR(IA,ID)=PROD+(1.-PROD)*Q(IA-1,ID-1)
             SLPTRY=RR(IA,ID)/IA
             IF(SLPTRY.GT.SLPMAX) SLPMAX=SLPTRY
           END DO
           SLOPE(ID)=SLPMAX
           NR(ID)=1
           DO IK=2,ID
             SLPMAX=0.
             PROD=(PFD2**IK)*CAPESS
             DO IA=1,R
               RTRY(IA)=PROD+(1.-PROD)*Q(IA-1,ID-IK)
               SLPTRY=RTRY(IA)/IA
               IF(SLPTRY.GT.SLPMAX) SLPMAX=SLPTRY
             END DO
             IF(SLPMAX.LT.SLOPE(ID)) THEN
               DO L=1,R
                 RR(L,ID)=RTRY(L)
               END DO
               SLOPE(ID)=SLPMAX
               NR(ID)=IK
             END IF
           END DO
           SLPMAX=0.
           DO IA=1,R
```

```
     Q(IA,ID)=RR(IA,ID)
     SLPTRY=Q(IA,ID)/IA
     IF(SLPTRY.GT.SLPMAX) SLPMAX=SLPTRY
    END DO
    NQ(ID)=0
    DO IK=1,ID
     SLPMAX=0.
     POW=PFD1**IK
     DO IA=1,R
      QTRY(IA)=POW*RR(IA,ID-IK)+(1.-POW)*Q(IA-1,ID-IK)
      SLPTRY=QTRY(IA)/IA
      IF(SLPTRY.GT.SLPMAX) SLPMAX=SLPTRY
     END DO
     IF(SLPMAX.LT.SLOPE(ID)) THEN
      DO L=1,R
       Q(L,ID)=QTRY(L)
      END DO
      SLOPE(ID)=SLPMAX
      NQ(ID)=IK
     END IF
    END DO
   END DO
c
c  RETURN THE SURVIVAL PROBABILITIES
c
   DO I=0,R
     DO J=0,S
      P(I,J)=1.-Q(I,J)
     END DO
   END DO
c
   DO ID=1,S
    NRR(ID)=ID-NQ(ID)
   END DO
c
   RETURN
   END
```

```
       SUBROUTINE SEQAT2(MAXR,MAXS,PFA,PFD1,PFD2,R,S,P)
c
c    Subroutine SEQAT2 solves for the probability of a target
c    surviving under known sequential attack and shoot, look, shoot,
c    defense
c
c    INPUT variables:
c
c     MAXR         INTEGER          The maximum allowable 'R'
c     MAXS         INTEGER          The maximum allowable 'S'
c     PFA          REAL             The probability that a
c                                    missile will fail to
c                                    destroy a target
c     PFD1         REAL             The probability that a
c                                    first salvo interceptor
c                                    will fail to destroy a
c                                    missile
c     PFD2         REAL             The probability that a
c                                    second salvo interceptor
c                                    will fail to destroy a
c                                    missile
c      R           INTEGER          The maximum number of
c                                    missiles that can attack
c                                    a target
c      S           INTEGER          The maximum number of
c                                    interceptors that can
c                                    defend a target
c
c   OUTPUT variable:
c
c     P(I,J)       REAL             The probability that a
c                                    target attacked by I
c                                    missiles and protected
c                                    by J interceptors will
c                                    survive
c
c
      INTEGER R,S
      REAL P(0:MAXR,0:MAXS)
c
c    LOCAL PROGRAM NOTES:
c
c    COMPUTES AN OPTIMAL PRIM READ DEFENSE WITH A
c    SLS CAPABILITY UNDER THE ASSUMPTION THAT THE
c    DEFENSE CAN DETERMINE THE SIZE OF THE ATTACK
c    AT THE MOMENT THAT THE ATTACK BEGINS.
c
c
c    DES(...) IS THE EXPECTED DESTRUCTION
c
   REAL SUR(100,100),DES(100,100),FRSLPE(100)
   INTEGER DEE(100,100),EEE(100,100)
c
c PARAMETER EPSI IS A TOLERANCE
```

```
C
      EPSI=0.000001
C
      SLOPE=1./R
C
C  IIPRIM WILL REMAIN 1 WHEN DES<=JA*SLOPE FOR ALL JA
C
      IIPRIM=1
      DO ID=1,S+1
         SUR(1,ID)=1.
         DES(1,ID)=0.
 1010    DO JA=2,R+1
            SUR(JA,ID)=PFA*SUR(JA-1,ID)
            DES(JA,ID)=1.-SUR(JA,ID)
C
C         IPRIM WILL BE SET TO 1 IF PRIMREAD OK AT JA,ID
C
            IPRIM=0
            DEE(JA,ID)=0
            EEE(JA,ID)=0
            SURTRY=SUR(JA,ID)
            DO LODEE=1,ID
               DO LOEEE=1,ID-LODEE+1
                  INDTEM=ID-LODEE-LOEEE+2
                  TRYSUR=(1.-PFD1**(LODEE-1))*SUR(JA-1,ID-LODEE+1)+
     1                    (PFD1**(LODEE-1))*
     1                    (1.-(PFD2**(LOEEE-1))*(1.-PFA))*
     1                    SUR(JA-1,INDTEM)
C
C TEST FOR UPDATE:  "THRIFTY ALLOCATION" BREAKS TIES
C                   IN FAVOR OF THE SMALLEST POTENTIAL
C                   ALLOCATIONS.  IF STILL TIED, IT PICKS
C                   THE SMALLEST INITIAL VOLLEY.
C
                  SRTRYN=SURTRY-EPSI
                  IF(TRYSUR.LT.SRTRYN) GO TO 1000
                  SRTRYP=SURTRY+EPSI
                  IF(TRYSUR.LE.SRTRYP) THEN
                    ISMINC=DEE(JA,ID)+EEE(JA,ID)+2
                    ISMTRY=LODEE+LOEEE
                    IF(ISMINC.LT.ISMTRY) GO TO 1000
                    IF(ISMINC.EQ.ISMTRY) THEN
                      IF(DEE(JA,ID).LE.LODEE-1) GO TO 1000
                    END IF
                  END IF
C  UPDATE IS NECESSARY
                     SUR(JA,ID)=TRYSUR
                     DES(JA,ID)=1.-SUR(JA,ID)
                     DEE(JA,ID)=LODEE-1
                     EEE(JA,ID)=LOEEE-1
                     SURTRY=TRYSUR
C  UPDATE UNECESSARY
 1000             CONTINUE
               END DO
            END DO
         END DO
      END DO
C
C DETERMINE WORST SLOPE FOR EACH DEFENSE SIZE
C
      DO ID=1,S+1
         PRSLPE(ID)=0.
         DO IA=2,R+1
            SLPTRY=DES(IA,ID)/(IA-1.)
            IF(SLPTRY.GT.PRSLPE(ID)) PRSLPE(ID)=SLPTRY
         END DO
      END DO
C
C  RETURN PIJ'S
C
      DO I=0,R
         DO J=0,S
             P(I,J)=1-DES(I+1,J+1)
         END DO
      END DO
C
      RETURN
      END
```

APPENDIX C

SUBROUTINES BG, YROBUST, AND XROBUST

```
c
        SUBROUTINE BG(XBG, YBG, VBG, R, S, P,
     1RV, A, INT, TARGETS,
     1MAT, MAXR, MAXS, MAXNTYPE, NTYPE,
     1VFRAC, NFRAC)
C
c       Subroutine BG finds and returns the solution(s) for the basic game
c
c*************************************************************************
c
c       INPUT variables:
c
c         R                  INTEGER              Maximum RV's at a single
c                                                   target
c         S                  INTEGER              Maximum interceptors at a
c                                                   single target
c         P(I,J)             REAL                 Probability of survival for
c                                                   a target attacked by I
c                                                   RV's and protected by J
c                                                   interceptors
c         RV                 INTEGER              The attack sizes
c         A                  INTEGER              The number of attack sizes
c         INT                INTEGER              Number of interceptors
c         TARGETS            INTEGER              Number of targets
c         MAT                INTEGER              Maximum permissable A
c         MAXR               INTEGER              Maximum permissable R
c         MAXS               INTEGER              Maximum permissable S
c         MAXNTYPE           INTEGER              Maximum permissable NTYPE
c         NTYPE              INTEGER              The number of target types
c         VFRAC(K)           REAL                 Fractional value of all
c                                                   type K targets
c         NFRAC(K)           REAL                 Fractional number of all
c                                                   type K targets
c
c*************************************************************************
c
c       Input variable type declaration
c
          INTEGER R, S, A, INT, TARGETS
          INTEGER MAT, MAXR, MAXS
          INTEGER RV(MAT), MAXNTYPE, NTYPE
          REAL P(0:MAXR,0:MAXS)
          REAL VFRAC(MAXNTYPE), NFRAC(MAXNTYPE)
c
c*************************************************************************
c
c       OUTPUT variables:
c
c         XBG(A, K, I)          REAL              Min-max strategy for the
c                                                   attacker for attack size
c                                                   A: fraction of K'th type
c                                                   targets with I RV's
c         YBG(A, K, J)          REAL              Min-max strategy for the
c                                                   defender for attack size
```

```
c                                                 A: fraction of K'th type
c                                                 targets with J interceptors
c       VBG(A)          REAL                    Game value associated with
c                                                 XBG and YBG
c
c*********************************************************************
c
c     Output variable type declaration
c
        REAL XBG(MAT,MAXNTYPE,0:MAXR), YBG(MAT,MAXNTYPE,0:MAXS)
        REAL VBG(MAT)

c
c*********************************************************************
c
c     Local variables needed for XMP.  See XMP Dictionary for
c     definitions.
c
c*********************************************************************
c
c     Integer parameters
c
        INTEGER MAXM,MAXN,MAXA,COLMAX,PICK
        PARAMETER (MAXM=550, MAXN=1000, MAXA=30000, COLMAX=550,
     1PICK=7)
        INTEGER LENI1, LENMI1, LENMR1, LENR1
        PARAMETER (LENI1=60000,LENMI1=9,LENMR1=8,LENR1=60000)
c
c     Double Precision arrays and variables
c
        DOUBLE PRECISION B(MAXM),BASCB(MAXM),BASLB(MAXM),BASUB(MAXM)
        DOUBLE PRECISION BLOW(1),BOUND,CANDA(COLMAX,PICK),CANDCJ(PICK)
        DOUBLE PRECISION CJ,COLA(COLMAX),LJ,MEMR(LENR1),UJ,UZERO(MAXM)
        DOUBLE PRECISION XBZERO(MAXM),YQ(MAXM),Z
c
c     Integer arrays and variables
c
        INTEGER BNDTYP,COLLEN,DFEASQ,DTERM,DUMBR,ERROR,FACTOR
        INTEGER IOERR,IOLOG,ITER,ITER1,ITER2,LOOK,M,MAPI(LENMI1)
        INTEGER MAPR(LENMR1),N,NTYPE2,PRINT,TERMIN
        INTEGER UNBDDQ,MEMI(LENI1),BASIS(MAXM),CAND(PICK)
        INTEGER CANDI(COLMAX,PICK),CANDL(PICK),COLI(COLMAX)
        INTEGER ROWTYP(MAXM), STATUS(MAXN)
c
c     I/O numbers for input, error output, and results output
c
        COMMON/IO_UNIT/IOIN,IOERR,IOLOG
c
c     Other common variables
c
        COMMON /XMPLEN/ LENI, LENMI, LENMR, LENR
        INTEGER LENI, LENMI, LENMR, LENR
c
c***************************************************************************
```

```
c
c                       Body of Program (SUBROUTINE BG)
c
c*******************************************************************************
c
c
          LENI-LENI1
          LENMI-LENMI1
          LENR-LENR1
          LENMR-LENMR1
          LOOK-200
          FACTOR-50
          PRINT-0
          BNDTYP-1
c
          CALL XMAPS(BNDTYP,IOERR,MAPI,MAPR,MAXA,MAXM,MAXN,MEMI,MEMR)
c
c       Prepare XMP to solve for the first attack size RV(1)
c       Set row type and right hand side for the inequality constraints
c
          N1-NTYPE*(R+1)
          DO 100 I-1, N1
              B(I)-0
              ROWTYP(I)-+1
100       CONTINUE
c
c       Set row type and right hand side for the equality constraints
c
          N2-NTYPE*(R+2)
          DO 110 I-N1+1, N2
              B(I)-1
              ROWTYP(I)-0
110       CONTINUE
          B(N2+1)-REAL(INT)/REAL(TARGETS)
          ROWTYP(N2+1)-0
c
c       Let N be the current number of variables; incremented by XADDAJ
c
          N-0
c
c       Construct the first NTYPE*(S+1) structural variables associated
c         with the defender's strategy
c
          DO 120 NT-1, NTYPE
              DO 121 J-0, S
                  COLLEN-0
                  DO 122 I-0, R
                      IF (P(I,J) .NE. 0) THEN
                          COLLEN-COLLEN+1
                          COLA(COLLEN)--P(I,J)*VFRAC(NT)
                          COLI(COLLEN)-I+1+(R+1)*(NT-1)
                      ENDIF
122               CONTINUE
                  COLLEN-COLLEN+1
```

```
c
c       Set the number of constraints
c
          M=N2+1
c
c       Construct the slack, surplus and artificial variables with XSLACK
c
          CALL XSLACK(B,BASCB,BASIS,BASLB,BASUB,BLOW,
     1                BNDTYP,BOUND,
     1                COLA,COLI,COLMAX,IOERR,
     1                M,MAPI,MAPR,MAXM,MAXN,MEMI,MEMR,
     1                N,ROWTYP,STATUS,UZERO,XBZERO,Z)
c
c       Solve the LP with the primal simplex method
c
          CALL XPRIML(B,BASCB,BASIS,BASLB,BASUB,BNDTYP,BOUND,
     1                CAND,CANDA,CANDCJ,CANDI,CANDL,
     1                COLA,COLI,COLMAX,
     1                FACTOR,IOERR,EOLOG,ITER1,ITER2,
     1                LOOK,M,MAPI,MAPR,MAXM,MAXN,MEMI,MEMR,
     1                N,NTYPE2,PICK,PRINT,STATUS,TERMIN,UNBDDQ,
     1                UZERO,XBZERO,YQ,Z)
c
c       Check if solution has been found
c
          IF (TERMIN .NE. 1) THEN
              WRITE (IOERR,'(A)') '1THERE IS AN ERROR IN THE FORMULATION
     1'
              WRITE (IOERR,'(A,I4)') ' TERMINATION CODE = ', TERMIN
              STOP
          ENDIF
c
c       Store in XBG, YBG, and VBG the solutions to the LP for RV(1)
c
          DO 200 NT=1, NTYPE
              DO 201 I= 0, R
                  XBG(1,NT,I)=UZERO(I+1+(NT-1)*(R+1))
201           CONTINUE
200       CONTINUE
          DO 210 I= 1, N2+1
              IF (BASIS(I) .LE. (S+1)*NTYPE) THEN
                  YBG(1,(BASIS(I)-1)/(S+1)+1,
     1            MOD(BASIS(I)-1,(S+1)))=XBZERO(I)
              ENDIF
210       CONTINUE
              VBG(1)=Z
c
c       Solve for the remaining RV's
c
          IF (A .EQ. 1) THEN
              RETURN
          ENDIF
c
          DO 300 I = 2, A
```

```
                  COLA(COLLEN)=1.0
                  COLI(COLLEN)=N1+NT
                  IF (J .NE. 0) THEN
                      COLLEN=COLLEN+1
                      COLA(COLLEN)=J*NFRAC(NT)
                      COLI(COLLEN)=N2+1
                  ENDIF
c
c           Call XADDAJ to enter the column for Yj (or variable J+1)
c
                  OBJ=0
                  CALL XADDAJ(OBJ,COLA,COLI,COLLEN,COLMAX,IOERR,
     1                        K,MAPI,MAPR,MEMI,MEMR,N)
121           CONTINUE
120       CONTINUE
c
c       Construct the columns for the 's' variables
c
          DO 130 NT=1, NTYPE
              DO 131 I=1,R+1
                  COLA(I)=1
                  COLI(I)=I+(NT-1)*(R+1)
131           CONTINUE
c
c           Call XADDAJ to enter the column for s (or variable S+2)
c
              COLLEN=R+1
              OBJ=1
              CALL XADDAJ(OBJ,COLA,COLI,COLLEN,COLMAX,IOERR,
     1                    K,MAPI,MAPR,MEMI,MEMR,N)
130       CONTINUE
c
c       Construct the column for the 't' variable
c
          COLLEN=0
          DO 140 NT=1, NTYPE
              DO 141 I=1,R
                  COLLEN=COLLEN+1
                  COLA(COLLEN)=-I*NFRAC(NT)
                  COLI(COLLEN)=I+1+(NT-1)*(R+1)
141           CONTINUE
140       CONTINUE
c
c       Call XADDAJ to enter the column for t (or variable S+3)
c
          OBJ=REAL(-RV(1))/REAL(TARGETS)
          CALL XADDAJ(OBJ,COLA,COLI,COLLEN,COLMAX,IOERR,
     1                K,MAPI,MAPR,MEMI,MEMR,N)
c
c       Start all the structural variables at their lower bounds
c
          DO 150 J=1, N
150           STATUS(J)=0
c
```

```
              J=NTYPE*(S+2)+1
              CJ=REAL(-RV(I))/REAL(TARGETS)
              CALL XCHGCJ(CJ,J,MAPI,MAPR,MEMI,MEMR)
c
              DO 310 I1=1,M
                  J=BASIS(I1)
                  CALL XGETAJ(CJ,COLA,COLI,COLLEN,COLMAX,
     1 IOERR,J,MAPI,MAPR,MEMI,MEMR)
                  BASCB(I1)=CJ
310           CONTINUE
c
          CALL XPRIML(B,BASCB,BASIS,BASLB,BASUB,BNDTYP,BOUND,
     1                      CAND,CANDA,CANDCJ,CANDI,CANDL,
     1                      COLA,COLI,COLMAX,
     1                      FACTOR,IOERR,EOLOG,ITER1,ITER2,
     1                      LOOK,M,MAPI,MAPR,MAXM,MAXN,MEMI,MEMR,
     1                      N,NTYPE2,PICK,PRINT,STATUS,TERMIN,UNBDDQ,
     1                      UZERO,XBZERO,YQ,Z)
c
c        Store in XBG, YBG, and VBG the solutions to the LP for RV(I)
c
              DO 320 NT=1, NTYPE
                  DO 321 I2= 0, R
                      XBG(I,NT,I2)=UZERO(I2+1+(NT-1)*(R+1))
321               CONTINUE
320           CONTINUE
              DO 330 I2= 1, N2+1
                  IF (BASIS(I2) .LE. (S+1)*NTYPE) THEN
                      YBG(I, (BASIS(I2)-1)/(S+1)+1,
     1MOD(BASIS(I2)-1,(S+1)))=XBZERO(I2)
                  ENDIF
330           CONTINUE

              VBG(I)=Z
300       CONTINUE
c
c
          RETURN
c
          END


c
c        Subroutines for changing an objective row coefficient
c


          SUBROUTINE XCHGCJ(CJ,J,MAPI,MAPR,MEMI,MEMR)
          COMMON/XMPLEN/LENI,LENMI,LENMR,LENR
          DOUBLE PRECISION CJ,MEMR(LENR)
          INTEGER MAPI(LENMI), MAPR(LENMR)
          INTEGER MEMI(LENI)
```

```
CALL XDATA5(CJ,J,MEMR(MAPR(3)),MEMI(MAPI(5)))
RETURN
END

SUBROUTINE XDATA5(CJ,J,PROFIT,MAXN)
DOUBLE PRECISION CJ,PROFIT(MAXN)
PROFIT(J)=CJ
RETURN
END
```

```
c
          SUBROUTINE YROBUST(YII, VBG, R, S, P,
     1RV, A, INT, TARGETS,
     1MAT, MAXR, MAXS, MAXNTYPE, NTYPE, NFRAC, VFRAC)
C
c       Subroutine ROBUST finds and returns the robust defender's strategy
c       over a range of attack sizes
c
c**************************   PRIMAL VERSION   ****************************
c
c************************************************************************
c
c       INPUT variables:
c
c         VBG(A)            REAL                 Game value associated with
c                                                  XBG and YBG
c         R                 INTEGER              Maximum RV's at a single
c                                                  target
c         S                 INTEGER              Maximum interceptors at a
c                                                  single target
c         P(I,J)            REAL                 Probability of survival for
c                                                  a target attacked by I RV's
c                                                  and protected by J
c                                                  interceptors
c         RV                INTEGER              The attack sizes over which
c                                                  the robust defense will be
c                                                  defined
c         A                 INTEGER              The number of attack sizes
c       INT                 INTEGER              Number of interceptors
c       TARGETS             INTEGER              Number of targets
c       MAT                 INTEGER              Maximum permissable A
c       MAXR                INTEGER              Maximum permissable R
c       MAXS                INTEGER              Maximum permissable S
c       MAXNTYPE            INTEGER              Maximum permissable NTYPE
c       NTYPE               INTEGER              Number of target types
c       NFRAC(K)            REAL                 The fraction of targets of
c                                                  type K
c         VFRAC(K)          REAL                 The fraction of the total
c                                                  value for type K targets
c
c************************************************************************
c
c       Input variable type declaration
c
          INTEGER R, S, A, INT, TARGETS
          INTEGER MAT, MAXR, MAXS, MAXNTYPE, NTYPE
          INTEGER RV(MAT)
          REAL P(0:MAXR,0:MAXS), VBG(MAT), NFRAC(MAXNTYPE)
          REAL VFRAC(MAXNTYPE)
c
c************************************************************************
c
c       OUTPUT variables:
```

```
C
C        YII(K,J)        REAL                 Robust strategy for the
C                                               defender over attack sizes
C                                               RV: fraction of type K
C                                               targets with J
C                                               interceptors
C
C*******************************************************************
C
C       Output variable type declaration
C
          REAL YII(MAXNTYPE,0:MAXS)
C
C*******************************************************************
C
C       Local variables needed for XMP.  See XMP Dictionary for
C       definitions.
C
C*******************************************************************
C
C       Integer parameters
C
          INTEGER MAXM,MAXN,MAXA,COLMAX,PICK
          PARAMETER (MAXM=1500, MAXN=1700, MAXA=20000)
          PARAMETER (COLMAX=100, PICK=7)
          INTEGER LENI1, LENMI1, LENMR1, LENR1
          PARAMETER (LENI1=100000,LENMI1=9,LENMR1=8,LENR1=50000)
C
C       Double Precision arrays and variables
C
          DOUBLE PRECISION B(MAXM),BASCB(MAXM),BASLB(MAXM),BASUB(MAXM)
          DOUBLE PRECISION BLOW(1),BOUND,CANDA(COLMAX,PICK),CANDCJ(PICK)
          DOUBLE PRECISION CJ,COLA(COLMAX),LJ,MEMR(LENR1),UJ,UZERO(MAXM)
          DOUBLE PRECISION XBZERO(MAXM),YQ(MAXM),Z
C
C       Integer arrays and variables
C
          INTEGER BNDTYP,COLLEN,DFEASQ,DTERM,DUMBR,ERROR,FACTOR
          INTEGER IOERR,IOLOG,ITER,ITER1,ITER2,LOOK,M,MAPI(LENMI1)
          INTEGER MAPR(LENMR1),N,NTYPE2,PRINT,TERMIN
          INTEGER UNBDDQ,MEMI(LENI1),BASIS(MAXM),CAND(PICK)
          INTEGER CANDI(COLMAX,PICK),CANDL(PICK),COLI(COLMAX)
          INTEGER ROWTYP(MAXM), STATUS(MAXN)
C
C       I/O numbers for in put, error output, and results output
C
          COMMON/IO_UNIT/IOIN,IOERR,IOLOG
C
C       Other common variables
C
          COMMON /XMPLEN/ LENI, LENMI, LENMR, LENR
          INTEGER LENI, LENMI, LENMR, LENR
C
C*******************************************************************
```

```
c
c                  Body of Program (SUBROUTINE YROBUST)
c
c***********************************************************************
c
c


         LENI=LENI1
         LENMI=LENMI1
         LENR=LENR1
         LENMR=LENMR1
         LOOK=200
         FACTOR=50
         PRINT=0
         BNDTYP=1
c
         CALL XMAPS(BNDTYP,IOERR,MAPI,MAPR,MAXA,MAXM,MAXN,MEMI,MEMR)
c
c       Set row type and right hand sides for the Z equality
c       constraints
c
         N1=(R+1)*NTYPE
         DO 100 I=1, N1
             B(I)=0
             ROWTYP(I)=0
100      CONTINUE
c
c       Set row type and right hand sides for the RO and ZST
c       inequality constraints
c
         N2=N1+A+A*NTYPE*(R+1)
         DO 110 I=N1+1, N2
             B(I)=0
             ROWTYP(I)=+1
110      CONTINUE
c
c       Set row type and right hand side for the Y equality constraints
c
         N3=N2+NTYPE
         DO 120 I=N2+1, N3
             B(I)=1
             ROWTYP(I)=0
120      CONTINUE
         B(N3+1)=REAL(INT)/TARGETS
         ROWTYP(N3+1)=0
c
c       Let N be the current number of variables; incremented by XADDAJ
c
         N=0
c
c       Construct the first NTYPE*S+1 structural variables associated
c         with the defender's strategy
c
```

```
          DO 130 NT=1, NTYPE
          DO 131 J=0, S
              COLLEN=0
              DO 132 I=0, R
                  IF (P(I,J) .NE. 0) THEN
                      COLLEN=COLLEN+1
                      COLA(COLLEN)=P(I,J)*VFRAC(NT)
                      COLI(COLLEN)=I+1+(R+1)*(NT-1)
                  ENDIF
132           CONTINUE
              COLLEN=COLLEN+1
              COLA(COLLEN)=1.0
              COLI(COLLEN)=N2+NT
              IF (J .NE. 0) THEN
                  COLLEN=COLLEN+1
                  COLA(COLLEN)=J*NFRAC(NT)
                  COLI(COLLEN)=N3+1
              ENDIF
c
c           Call XADDAJ to enter the column for Yj (or variable J+1)
c
              OBJ=0
              CALL XADDAJ(OBJ,COLA,COLI,COLLEN,COLMAX,IOERR,
     1                    K,MAPI,MAPR,MEMI,MEMR,N)
131       CONTINUE
130       CONTINUE
c
c       Construct the columns for the z variables
c
          DO 140 NT=1,NTYPE
              DO 141 I=1,R+1
                  COLA(1)=-1.0
                  COLI(1)=I+(NT-1)*(R+1)
                  DO 142 I1=1,A
                      COLA(I1+1)=-1.0
                      COLI(I1+1)=N1+A+A*(R+1)*(NT-1)+(I1-1)*(R+1)+I
142               CONTINUE
c
c               Call XADDAJ to enter the column for z's
c
                  COLLEN=A+1
                  OBJ=0

                  CALL XADDAJ(OBJ,COLA,COLI,COLLEN,COLMAX,IOERR,
     1                    K,MAPI,MAPR,MEMI,MEMR,N)
141           CONTINUE
140       CONTINUE
c
c       Construct the column for RO
c
          DO 150 I=1,A
              COLA(I)=VBG(I)
              COLI(I)=N1+I
150       CONTINUE
```

```
          COLLEN-A
          OBJ-1
          CALL XADDAJ(OBJ,COLA,COLI,COLLEN,COLMAX,IOERR,
     1                K,MAPI,MAPR,MEMI,MEMR,N)
c
c       Construct the column for the 's' variables
c
          DO 160 NT-1, NTYPE
              DO 161 I-1,A
                  COLA(1)--1
                  COLI(1)-N1+I
                  DO 162 I1-1, R+1
                      COLA(I1+1)-1
                      COLI(I1+1)-N1+A+A*(R+1)*(NT-1)+(I-1)*(R+1)+I1
162               CONTINUE
                  COLLEN-R+2
                  OBJ-0
                  CALL XADDAJ(OBJ,COLA,COLI,COLLEN,COLMAX,IOERR,
     1                    K,MAPI,MAPR,MEMI,MEMR,N)
161           CONTINUE
160       CONTINUE
c
c       Construct the column for the 't' variables
c
          DO 170 I-1,A
              COLA(1)-REAL(RV(I))/REAL(TARGETS)
              COLI(1)-N1+I
              COLLEN-1
              DO 171 NT-1,NTYPE
                  DO 172 I1-1, R
                      COLLEN-COLLEN+1
                      COLA(COLLEN)--I1*NFRAC(NT)
                      COLI(COLLEN)-N1+A+(NT-1)*A*(R+1)+(I-1)*(R+1)+I1+1
172               CONTINUE
171           CONTINUE
              OBJ-0
c
c           Call XADDAJ to enter the column for t (or variable S+3)
c
              CALL XADDAJ(OBJ,COLA,COLI,COLLEN,COLMAX,IOERR,
     1                    K,MAPI,MAPR,MEMI,MEMR,N)
170       CONTINUE
c
c       Start all the structural variables at their lower bounds
c
          DO 180 J-1, N
180           STATUS(J)-0
c
c       Set the number of constraints
c
          M-N3+1
c
c       Construct the slack, surplus and artificial variables with XSLACK
```

```
c
          CALL XSLACK(B,BASCB,BASIS,BASLB,BASUB,BLOW,
     1                BNDTYP,BOUND,
     1                COLA,COLI,COLMAX,IOERR,
     1                M,MAPI,MAPR,MAXM,MAXN,MEMI,MEMR,
     1                N,ROWTYP,STATUS,UZERO,XBZERO,Z)
c
c       Solve the LP with the primal simplex method
c
          CALL  XPRIML(B,BASCB,BASIS,BASLB,BASUB,BNDTYP,BOUND,
     1                 CAND,CANDA,CANDCJ,CANDI,CANDL,
     1                 COLA,COLI,COLMAX,
     1                 FACTOR,IOERR,IOLOG,ITER1,ITER2,
     1                 LOOK,
     1                 M,MAPI,MAPR,MAXM,MAXN,MEMI,MEMR,
     1                 N,NTYPE2,PICK,PRINT,STATUS,
     1                 TERMIN,UNBDDQ,
     1                 UZERO,XBZERO,YQ,Z)
c
c       Check if solution has been found
c
          IF (TERMIN .NE. 1) THEN
              WRITE (IOERR,'(A)') '1THERE IS AN ERROR IN THE FORMULATION
     1'
              WRITE (IOERR,'(A,I4)') ' TERMINATION CODE - ', TERMIN
              STOP
          ENDIF
c
c       Store in YII the solution to the LP
c
          DO 200 I-1,NTYPE
              DO 201 J-0, S
201               YII(I,J)-0
200           CONTINUE
          DO 210 I- 1, M
              IF (BASIS(I) .LE. (S+1)*NTYPE) THEN
                  YII((BASIS(I)-1)/(S+1)+1,MOD(BASIS(I)-1,(S+1)))
     1-XBZERO(I)
              ENDIF
210       CONTINUE
c
          RETURN
c
          END
```

```
c
        SUBROUTINE XROBUST(XII, YII, VII, R, S, P,
     1RV, A, INT, TARGETS,
     1MAT, MAXR, MAXS, MAXNTYPE, NTYPE, NFRAC, VFRAC)
c
c       Subroutine XROBUST uses the robust defense to find the optimal
c       attack against it and the resultant expected target survival
c       rate for each specified attack size.
c
c************************************************************************
c
c       INPUT variables:
c
c         YII(K,J)          REAL              The robust defense found by
c                                               YROBUST: fraction of type
c                                               K targets protected by J
c                                               interceptors
c         R                 INTEGER           Maximum RV's at a single
c                                               target
c         S                 INTEGER           Maximum interceptors at a
c                                               single target
c         P(I,J)            REAL              Probability of survival for a
c                                               target attacked by I RV's
c                                               and protected by J
c                                               interceptors
c         RV                INTEGER           The attack sizes
c         A                 INTEGER           The number of attack sizes
c         INT               INTEGER           Number of interceptors
c         TARGETS           INTEGER           Number of targets
c         MAT               INTEGER           Maximum permissable A
c         MAXR              INTEGER           Maximum permissable R
c         MAXS              INTEGER           Maximum permissable S
c         MAXNTYPE          INTEGER           Maximum permissable NTYPE
c         NTYPE             INTEGER           The number of target types
c         NFRAC(K)          REAL              The fraction of all targets
c                                               that are of type K
c         VFRAC(K)          REAL              The fraction of total target
c                                               value that are of type K
c
c************************************************************************
c
c       Input variable type declaration
c
          INTEGER R, S, A, INT, TARGETS
          INTEGER MAT, MAXR, MAXS, MAXNTYPE, NTYPE
          INTEGER RV(MAT)
          REAL P(0:MAXR,0:MAXS), YII(MAXNTYPE, 0:MAXS), NFRAC(MAXNTYPE)
          REAL VFRAC(MAXNTYPE)
c
c************************************************************************
c
c       OUTPUT variables:
c
c         XII(A, K, I)      REAL              The attacker's optimal
```

```
c                                                  strategy against YII for
c                                                  attack size A:  fraction
c                                                  of type K targets assigned
c                                                  I RV's
c        VII(A)           REAL                   The expected target survival
c                                                  rate associated with XII
c                                                  and YII
c
c*********************************************************************
c
c      Output variable type declaration
c
        REAL XII(MAT,MAXNTYPE,0:MAXR), VII(MAT)
c
c*********************************************************************
c
c      Local variables needed for XMP.  See XMP Dictionary for
c      definitions.
c
c*********************************************************************
c
c      Integer parameters
c
        INTEGER MAXM,MAXN,MAXA,COLMAX,PICK
        PARAMETER (MAXM=15, MAXN=100, MAXA=350, COLMAX=3, PICK=6)
        INTEGER LENI1, LENMI1, LENMR1, LENR1
        PARAMETER (LENI1=1000,LENMI1=9,LENMR1=8,LENR1=1000)
c
c      Double Precision arrays and variables
c
        DOUBLE PRECISION B(MAXM),BASCB(MAXM),BASLB(MAXM),BASUB(MAXM)
        DOUBLE PRECISION BLOW(1),BOUND,CANDA(COLMAX,PICK),CANDCJ(PICK)
        DOUBLE PRECISION CJ,COLA(COLMAX),LJ,MEMR(LENR1),UJ,UZERO(MAXM)
        DOUBLE PRECISION XBZERO(MAXM),YQ(MAXM),Z
c
c      Integer arrays and variables
c
        INTEGER BNDTYP,COLLEN,DFEASQ,DTERM,DUMBR,ERROR,FACTOR
        INTEGER IOERR,IOLOG,ITER,ITER1,ITER2,LOOK,M,MAPI(LENMI1)
        INTEGER MAPR(LENMR1),N,NTYPE2,PRINT,TERMIN
        INTEGER UNBDDQ,MEMI(LENI1),BASIS(MAXM),CAND(PICK)
        INTEGER CANDI(COLMAX,PICK),CANDL(PICK),COLI(COLMAX)
        INTEGER ROWTYP(MAXM), STATUS(MAXN)
c
c      I/O numbers for input, error output, and results output
c
        COMMON/IO_UNIT/IOIN,IOERR,IOLOG
c
c      Other common variables
c
        COMMON/XMPLEN/LENI, LENMI, LENMR, LENR
        INTEGER LENI, LENMI, LENMR, LENR
c
c*********************************************************************
```

```
c
c                  Body of Program (SUBROUTINE XROBUST)
c
c*************************************************************************
c
c

          LENI=LENI1
          LENMI=LENMI1
          LENR=LENR1
          LENMR=LENMR1
          LOOK=50
          FACTOR=50
          PRINT=0
          BNDTYP=1
c
          CALL XMAPS(BNDTYP,IOERR,MAPI,MAPR,MAXA,MAXM,MAXN,MEMI,MEMR)
c
c       Set row type and right hand side for the equality constraints
c
          DO 100 NT=1, NTYPE
              B(NT)=1
              ROWTYP(NT)=0
100       CONTINUE
          B(NTYPE+1)=REAL(RV(1))/REAL(TARGETS)
          ROWTYP(NTYPE+1)=0
c
c       Let N be the current number of variables; incremented by XADDAJ
c
          N=0
c
c       Construct the NTYPE*(R+1) structural variables associated with
c         the attacker's strategy
c
          DO 110 NT=1, NTYPE
              DO 111 I=0, R
                  COLLEN=1
                  COLA(1)=1
                  COLI(1)=NT
                  IF (I .NE. 0) THEN
                      COLLEN=2
                      COLA(2)=I*NFRAC(NT)
                      COLI(2)=NTYPE+1
                  ENDIF
                  OBJ=0
                  DO 112 J=0, S
112                   OBJ=-P(I,J)*VFRAC(NT)*YII(NT,J)+OBJ
                      CALL XADDAJ(OBJ,COLA,COLI,COLLEN,COLMAX,
     1                   IOERR,K,MAPI,MAPR,MEMI,MEMR,N)
111           CONTINUE
110       CONTINUE
c
c       Start all the structural variables at their lower bounds
c
```

```
          DO 120 J-1, N
120           STATUS(J)-0
C
C
C       Set the number of constraints
C
          M-NTYPE+1
C
C       Construct the slack, surplus and artificial variables with XSLACK
C
          CALL XSLACK(B,BASCB,BASIS,BASLB,BASUB,BLOW,
     1                BNDTYP,BOUND,
     1                COLA,COLI,COLMAX,IOERR,
     1                M,MAPI,MAPR,MAXM,MAXN,MEMI,MEMR,
     1                N,ROWTYP,STATUS,UZERO,XBZERO,Z)
C
C       Solve the LP with the primal simplex method
C
          CALL XPRIML(B,BASCB,BASIS,BASLB,BASUB,BNDTYP,BOUND,
     1                CAND,CANDA,CANDCJ,CANDI,CANDL,
     1                COLA,COLI,COLMAX,
     1                FACTOR,IOERR,EOLOG,ITER1,ITER2,
     1                LOOK,M,MAPI,MAPR,MAXM,MAXN,MEMI,MEMR,
     1                N,NTYPE2,PICK,PRINT,STATUS,TERMIN,UNBDDQ,
     1                UZERO,XBZERO,YQ,Z)
C
C       Check if solution has been found
C
          IF (TERMIN .NE. 1) THEN
              WRITE (IOERR,'(A)') '1THERE IS AN ERROR IN THE FORMULATION
     1'
              WRITE (IOERR,'(A,I4)') ' TERMINATION CODE - ', TERMIN
              STOP
          ENDIF
C
C       Store in XII and VII the solutions to the LP for RV(1)
C
          DO 200 I- 1, NTYPE+1
              IF (BASIS(I) .LE. NTYPE*(R+1)) THEN
                  XII(1, (BASIS(I)-1)/(S+1)+1,
     1            MOD(BASIS(I)-1,(S+1)))-XBZERO(I)
              ENDIF
200       CONTINUE
              VII(1)--Z
C
C       Solve for the remaining RV's
C
          IF (A .EQ. 1) THEN
              RETURN
          ENDIF
C
          DO 300 I - 2, A
C
              B(NTYPE+1)-REAL(RV(I))/REAL(TARGETS)
              CALL XBCOMP(B,BASCB,BNDTYP,BOUND,
     1                    COLA,COLI,COLMAX,IOERR,
     1                    M,MAPI,MAPR,MAXM,MAXN,MEMI,MEMR,N,
     1                    STATUS,XBZERO,Z)
C
              CALL XPRIML(B,BASCB,BASIS,BASLB,BASUB,BNDTYP,BOUND,
     1                    CAND,CANDA,CANDCJ,CANDI,CANDL,
     1                    COLA,COLI,COLMAX,
     1                    FACTOR,IOERR,EOLOG,ITER1,ITER2,
     1                    LOOK,M,MAPI,MAPR,MAXM,MAXN,MEMI,MEMR,
     1                    N,NTYPE2,PICK,PRINT,STATUS,TERMIN,UNBDDQ,
     1                    UZERO,XBZERO,YQ,Z)
C
C        Store in XII and VII the solutions to the LP for RV(I)
C
              DO 301 I1 = 1, M
                IF (BASIS(I1) .LE. NTYPE*(R+1)) THEN
                  XII(I, (BASIS(I1)-1)/(S+1)+1,
     1            MOD(BASIS(I1)-1,(S+1)))=XBZERO(I1)
                ENDIF
301           CONTINUE
              VII(I)=-Z
300           CONTINUE
C
           RETURN
C
           END
```

APPENDIX D

SUBROUTINES SUMMARY , YRPRINT, STPRINT, AND VPRINT

```
                SUBROUTINE SUMMARY(NATTYPE,NDFTYPE,ANSWER,MINRV,MAXRV,
     1                             INCRV,INT,TARGETS,R,S,PFA,PFD,PFD1,
     1                             PFD2,OUT,MAXNTYPE,NTYPE,VTYPE,NTAR)
c
          INTEGER NATTYPE,NDFTYPE,MINRV,MAXRV,INCRV,INT,TARGETS,R,S
          INTEGER OUT,MAXNTYPE,NTYPE
          INTEGER NTAR(MAXNTYPE)
          REAL PFA,PFD,PFD1,PFD2,VTYPE(MAXNTYPE)
          CHARACTER*1 ANSWER,TITLE*50
c
          WRITE(OUT,'(A)') 'OTHE PARAMETERS OF THIS PREALLOCATED PREFERE
     1NTIAL DEFENSE GAME'
          WRITE(OUT,1000) '0'
          TITLE='0 THE ATTACK METHODOLOGY'
          IF (NATTYPE .EQ. 1) THEN
              WRITE(OUT,1001) TITLE, 'SIMULTANEOUS'
          ELSE
              WRITE(OUT,1001) TITLE, 'SEQUENTIAL'
              IF (ANSWER .EQ. 'Y') THEN
                  WRITE(OUT,1001) '   ', '  WITH ATTACK SIZE AT'
                  WRITE(OUT,1001) '   ', '  A TARGET KNOWN TO'
                  WRITE(OUT,1001) '   ', '  THE DEFENDER'
              ELSE
                  WRITE(OUT,1001) '   ', '  WITH ATTACK SIZE AT'
                  WRITE(OUT,1001) '   ', '  A TARGET UNKNOWN TO'
                  WRITE(OUT,1001) '   ', '  THE DEFENDER'
              ENDIF
          ENDIF
          TITLE='  THE DEFENSE METHODOLOGY'
          IF (NDFTYPE .EQ. 1) THEN
              WRITE(OUT,1001) TITLE, 'ONE SHOT'
          ELSE
              WRITE(OUT,1001) TITLE, 'SHOOT LOOK SHOOT'
          ENDIF
c
          TITLE='0 THE FAILURE RATE OF RV''S'
          WRITE(OUT,1002) TITLE, PFA
          IF (NDFTYPE .EQ. 1) THEN
              TITLE='  THE FAILURE RATE OF THE INTERCEPTORS'
              WRITE(OUT,1002) TITLE, PFD
          ELSE
              TITLE='  THE FAILURE RATE OF THE FIRST SALVO'
              WRITE(OUT,1001) TITLE, ' '
              TITLE='       INTERCEPTORS'
              WRITE(OUT,1002) TITLE, PFD1
              TITLE='  THE FAILURE RATE OF THE SECOND SALVO'
              WRITE(OUT,1001) TITLE, ' '
              TITLE='       INTERCEPTORS'
              WRITE(OUT,1002) TITLE, PFD2
          ENDIF
c
          TITLE='0 MAXIMUM NUMBER OF RV''S ATTACKING A '
          WRITE(OUT,1001) TITLE, ' '
          TITLE='       SINGLE TARGET'
          WRITE(OUT,1003) TITLE, R
          TITLE='  MAXIMUM NUMBER OF INTERCPTORS DEFENDING A '
          WRITE(OUT,1001) TITLE, ' '
          TITLE='       SINGLE TARGET'
          WRITE(OUT,1003) TITLE, S
c
          TITLE='0 THE MINIMUM NUMBER OF RV''S'
          WRITE(OUT,1003) TITLE, MINRV
          TITLE='  THE MAXIMUM NUMBER OF RV''S'
          WRITE(OUT,1003) TITLE, MAXRV
          TITLE='  THE ATTACK SIZE INCREMENT'
          WRITE(OUT,1003) TITLE, INCRV
          TITLE='  THE NUMBER OF INTERCEPTORS'
          WRITE(OUT,1003) TITLE, INT
          TITLE='  THE TOTAL NUMBER OF TARGETS'
          WRITE(OUT,1003) TITLE, TARGETS
          TITLE='0 THE NUMBER OF TARGET TYPES'
          WRITE(OUT,1003) TITLE, NTYPE
          DO 100 I=1, NTYPE

              WRITE(OUT,1004) '        TARGET TYPE ', I,':'
              TITLE='             NUMBER OF TARGETS'
              WRITE(OUT,1003) TITLE, NTAR(I)
              TITLE='             RELATIVE VALUE'
              WRITE(OUT,1002) TITLE, VTYPE(I)
100       CONTINUE
              WRITE(OUT,1000) '0'
c
1000       FORMAT(A,72('*'))
1001       FORMAT(A50, A)
1002       FORMAT(A50, F5.3)
1003       FORMAT(A50, I5)
1004       FORMAT(A,I2,A)
c

          RETURN
          END


                SUBROUTINE YRPRINT(YR, MINRRV, MAXRRV, MAXS, N, OUT,
     1                                   MAXNTYPE, NTYPE)
c
          INTEGER MINRRV, MAXRRV, MAXS, MAXNTYPE, NTYPE
          REAL YR(MAXNTYPE,0:MAXS)
          INTEGER OUT, N1, N
c
          WRITE(OUT, 1000) '1THE ROBUST DEFENSE STRATEGY FOR RV RANGE '
     1, MINRRV, ' TO ', MAXRRV, ' :'
          WRITE(OUT,1001) '0'
          DO 100 NT=1,NTYPE
              WRITE(OUT,1002) '0','TARGET TYPE ',NT
              WRITE(OUT,1005) (I,I=0,9)
              WRITE(OUT,1003) ' '
          N1=N/10
          IF (N1 .EQ. 0) THEN
              WRITE(OUT,1004) (YR(NT,I), I=0,N)
          ELSE
              WRITE(OUT,1004) (YR(NT,I), I=0,9)
              DO 101 I1=1, N1
                  IF (I1 .EQ. N1) THEN
                      WRITE(OUT,1004) (YR(NT,I), I=I1*10, N)
                  ELSE
                      WRITE(OUT,1004) (YR(NT,I), I=I1*10,I1*10+9)
                  ENDIF
101           CONTINUE
          ENDIF
          WRITE(OUT,1003) ' '
100   CONTINUE
      WRITE(OUT,1001) '0'
c
1000  FORMAT(A,I5,A,I5,A)
1001  FORMAT(A,106('*'))
1002  FORMAT(A,47X,A,I2)
1003  FORMAT(A,7X,93('-'))
1004  FORMAT(8X,' :',10(F7.4,' :'))
1005  FORMAT(13X,10(I1,8X))
c
      RETURN
      END


            SUBROUTINE STPRINT(STNAME,STBG,RV,A,N,OUT,MAXNAT,MAXN,
     1                            MAXNTYPE,NTYPE,VFRAC,NFRAC)
c
      INTEGER MAXNAT,MAXN,MAXNTYPE,NTYPE
      CHARACTER*70 STNAME
      REAL STBG(MAXNAT,MAXNTYPE,0:MAXN), NFRAC(MAXNTYPE)
      REAL VFRAC(MAXNTYPE)
      INTEGER RV(MAXNAT), A, N, OUT
c
      WRITE(OUT,1000) STNAME
      WRITE(OUT,1001) '0'
      DO 100 NT=1, NTYPE
          WRITE(OUT,1000) ' '
          WRITE(OUT,1002) '0', 'TARGET TYPE', NT,
     1':   ', 100*NFRAC(NT),'% OF TOTAL TARGETS, WITH ', VFRAC(NT)*100,
     1'% OF TOTAL VALUE'
          IF (NT .EQ. 1) THEN
              WRITE(OUT,1007) '  ATTACK SIZE',(I, I=0,9)
          ELSE
              WRITE(OUT,1007) '             ',(I, I=0,9)
          ENDIF
          WRITE(OUT,1003) ' '
          DO 101 K = 1, A
              N1=N/10
              IF (N1 .EQ. 0) THEN
```

```
                      WRITE(OUT,1004) RV(K), (STBG(K,NT,I), I=0,N)
                  ELSE
                      WRITE(OUT,1004) RV(K), (STBG(K,NT,I), I=0,9)
                      DO 102 I1=1, N1
                          IF (I1 .EQ. N1) THEN
                              WRITE(OUT,1005) (STBG(K,NT,I), I=I1*10, N)
                          ELSE
                              WRITE(OUT,1005) (STBG(K,NT,I), I=I1*10,
     1 I1*10+9)
                          ENDIF
102                   CONTINUE
                  ENDIF
                  IF (K .NE. A) THEN
                      WRITE(OUT,1006)
                  ELSE
                      WRITE(OUT,1003) ' '
                  ENDIF
101           CONTINUE
100       CONTINUE
          WRITE(OUT,1000) ' '
          WRITE(OUT,1001) '0'
c
1000      FORMAT(A)
1001      FORMAT(A,106('*'))
1002      FORMAT(A,17X,A12,2X,I3,A,F6.2,A,F6.2,A)
1003      FORMAT(A,106('-'))
1004      FORMAT(5X, I5, 4X,' :',10(F7.4,' :'))
1005      FORMAT(14X,' :',10(F7.4,' :'))
1006      FORMAT(14X,93('-'))
1007      FORMAT(A,7X,10(I1,8X))
c
          RETURN
          END




                SUBROUTINE VPRINT(VNAME, VG, RV, A, OUT, MAXNAT)
c
c
          INTEGER MAXNAT
          CHARACTER*70 VNAME
          REAL VG(MAXNAT)
          INTEGER RV(MAXNAT), A, OUT
c
          WRITE(OUT,1000) VNAME
          WRITE(OUT,1001) '0'
          WRITE(OUT,1000) '0 ATTACK SIZE   '
          WRITE(OUT,1002) ' '
          DO 100 K= 1, A
              WRITE(OUT,1003) RV(K), VG(K)
100       CONTINUE
          WRITE(OUT,1002) ' '
```

```
            WRITE(OUT,1001) '0'
c
1000        FORMAT(A)
1001        FORMAT(A,29('*'))
1002        FORMAT(A,29('-'))
1003        FORMAT(5X, I5, 5X,':',F11.4)
c
            RETURN
            END
```

APPENDIX E

SUBROUTINES ALYRPRINT, ALPRINT, ALVPRINT, AND RVINTCOUNT

```
                SUBROUTINE ALYRPRINT(YR, MINRRV, MAXRRV, MAXS, N, OUT,
     1                               MAXNTYPE, NTYPE, NFRAC,TARGETS)
C
          INTEGER MINRRV, MAXRRV, MAXS, MAXNTYPE, NTYPE
          REAL YR(MAXNTYPE,0:MAXS),NFRAC(MAXNTYPE)
          INTEGER OUT, N1, N, TARGETS
C
          WRITE(OUT,1000) '1THE ROBUST DEFENSE ALLOCATION FOR RV RANGE '
     1, MINRRV, ' TO ', MAXRRV, ' :'
          WRITE(OUT,1001) '0'
          DO 100 NT-1,NTYPE
              WRITE(OUT,1002) '0','TARGET TYPE ',NT
              WRITE(OUT,1005) (I,I-0,9)
              WRITE(OUT,1003) ' '
              N1-N/10
              IF (N1 .EQ. 0) THEN
                  WRITE(OUT,1004) (YR(NT,I)*NFRAC(NT)*TARGETS , I-0,N)
              ELSE
                  WRITE(OUT,1004) (YR(NT,I)*NFRAC(NT)*TARGETS , I-0,9)
                  DO 101 I1-1, N1
                      IF (I1 .EQ. N1) THEN
                          WRITE(OUT,1004) (YR(NT,I)*NFRAC(NT)*TARGETS,
     1 I-I1*10, N)
                      ELSE
                          WRITE(OUT,1004) (YR(NT,I)*NFRAC(NT)*TARGETS,
     1 I-I1*10,I1*10+9)
                      ENDIF
101               CONTINUE
              ENDIF
              WRITE(OUT,1003) ' '
100       CONTINUE
          WRITE(OUT,1001) '0'
C
1000      FORMAT(A,I5,A,I5,A)
1001      FORMAT(A,106('*'))
1002      FORMAT(A,46X,A,I2)
1003      FORMAT(A,7X,93('-'))
1004      FORMAT(8X,' :',10(F7.1,' :'))
1005      FORMAT(13X,10(I1,8X))
C
          RETURN
          END


                SUBROUTINE ALPRINT(STNAME,STBG,RV,A,N,OUT,MAXNAT,MAXN,
     1                             MAXNTYPE,NTYPE,VFRAC,NFRAC,TARGETS)
C
          INTEGER MAXNAT,MAXN,MAXNTYPE,NTYPE
          CHARACTER*70 STNAME
          REAL STBG(MAXNAT,MAXNTYPE,0:MAXN), NFRAC(MAXNTYPE)
          REAL VFRAC(MAXNTYPE)
          INTEGER RV(MAXNAT), A, N, OUT,TARGETS
c
          WRITE(OUT,1000) STNAME
          WRITE(OUT,1001) '0'
          DO 100 K=1,A
              WRITE(OUT,1000) ' '
              WRITE(OUT,1002) '0', 'ATTACK SIZE =', RV(K)
              IF (K .EQ. 1) THEN
                  WRITE(OUT,1007) '  TARGET TYPE', (I, I=0,9)
              ELSE
                  WRITE(OUT,1007) '              ', (I, I=0,9)
              ENDIF
              WRITE(OUT,1003) ' '
              DO 101 NT=1,NTYPE
                  N1=N/10
                  IF (N1 .EQ. 0) THEN
                      WRITE(OUT,1004) NT,
     1 (STBG(K,NT,I)*NFRAC(NT)*TARGETS, I=0,N)
                  ELSE
                      WRITE(OUT,1004) NT,
     1 (STBG(K,NT,I)*NFRAC(NT)*TARGETS, I=0,9)
                      DO 102 I1=1, N1
                          IF (I1 .EQ. N1) THEN
                              WRITE(OUT,1005)
     1 (STBG(K,NT,I)*NFRAC(NT)*TARGETS, I=I1*10, N)
                          ELSE
                              WRITE(OUT,1005)
     1 (STBG(K,NT,I)*NFRAC(NT)*TARGETS, I=I1*10,I1*10+9)
                          ENDIF
102                   CONTINUE
                  ENDIF
                  IF (NT .NE. NTYPE) THEN
                      WRITE(OUT,1006)
                  ELSE
                      WRITE(OUT,1003) ' '
                  ENDIF
101           CONTINUE
100       CONTINUE
          WRITE(OUT,1000) ' '
          WRITE(OUT,1001) '0'
c
1000      FORMAT(A)
1001      FORMAT(A,106('*'))
1002      FORMAT(A,50X,A13,I6)
1003      FORMAT(A,106('-'))
1004      FORMAT(5X, I3, 6X,' :',10(F7.1,' :'))
1005      FORMAT(14X,' :',10(F7.1,' :'))
1006      FORMAT(14X,93('-'))
1007      FORMAT(A,7X,10(I1,8X))
c
          RETURN
          END
```

```
                SUBROUTINE ALVPRINT(VNAME, VG, RV, A, OUT, MAXNAT,
     1                               MAXNTYPE,NTYPE,TS)
C
C
          INTEGER MAXNAT,MAXNTYPE,NTYPE
          CHARACTER*70 VNAME, CNAME, BLANK
          REAL VG(MAXNAT), TS(MAXNAT,MAXNTYPE)
          INTEGER RV(MAXNAT), A, OUT
          CHARACTER*10 DASH(20), STAR(20)
C
          BLANK='                                                        '
          WRITE(OUT,1000) VNAME
C
          DO 200 NT=1, NTYPE
              STAR(NT)='**********'
              DASH(NT)='----------'
200       CONTINUE
C
          CNAME=BLANK(1:NTYPE*5)//'TARGET TYPE'
          WRITE(OUT,1001) '0',(STAR(NT), NT=1,NTYPE)
          WRITE(OUT,1002) '0', CNAME
          WRITE(OUT,1003) 'ATTACK SIZE', (I, I=1,NTYPE)
          WRITE(OUT,1004) ' ',(DASH(NT), NT=1,NTYPE)
          DO 100 K= 1, A
              WRITE(OUT,1005) RV(K),(TS(K,NT), NT=1,NTYPE)
100       CONTINUE
          WRITE(OUT,1004) ' ',(DASH(NT), NT=1,NTYPE)
          WRITE(OUT,1001) '0',(STAR(NT), NT=1,NTYPE)
C
1000      FORMAT(A)
1001      FORMAT(A,16('*'),10A)
1002      FORMAT(A,10X,A)
1003      FORMAT(2X,A11,2X,':   ',10(I4,6X))
1004      FORMAT(A,16('-'),10A)
1005      FORMAT(5X,I5,5X,':   ',10(F7.2,3X))
C
          RETURN
          END
```

```
                    SUBROUTINE RVINTCOUNT(VNAME, TM, RV, A, OUT,
     1                                    MAXNAT,MAXNTYPE,NTYPE,ROB)
c
c
          INTEGER MAXNAT,MAXNTYPE,NTYPE
          CHARACTER*70 VNAME, CNAME, BLANK
          INTEGER RV(MAXNAT), A, OUT, TM(MAXNAT,MAXNTYPE), N
          CHARACTER*10 DASH(20), STAR(20)
          LOGICAL ROB

c
          BLANK='                                                                      '
          WRITE(OUT,1000) VNAME
          DO 100 NT=1, NTYPE
              STAR(NT)='**********'
              DASH(NT)='----------'
100       CONTINUE
          CNAME=BLANK(1:NTYPE*5)//'TARGET TYPE'
          WRITE(OUT,1001) '0',(STAR(NT), NT=1,NTYPE)
          WRITE(OUT,1002) '0', CNAME
          WRITE(OUT,1003) 'ATTACK SIZE', (I, I=1,NTYPE)
          WRITE(OUT,1004) ' ',(DASH(NT), NT=1,NTYPE)
          IF (ROB .EQ. .FALSE.) THEN
              DO 50 K= 1, A
                  WRITE(OUT,1005) RV(K),(TM(K,NT), NT=1,NTYPE)
50            CONTINUE
          ELSE
              WRITE(OUT,1006) 'N/A', (TM(1,NT), NT=1,NTYPE)

          ENDIF
              WRITE(OUT,1004) ' ',(DASH(NT), NT=1,NTYPE)
              WRITE(OUT,1001) '0',(STAR(NT), NT=1,NTYPE)
c
1000      FORMAT(A)
1001      FORMAT(A,16('*'),10A)
1002      FORMAT(A,10X,A)
1003      FORMAT(2X,A11,2X,':  ',10(I4,6X))
1004      FORMAT(A,16('-'),10A)
1005      FORMAT(5X, I5, 5X,':  ',10(I6,4X))
1006      FORMAT(6X, A3, 6X,':  ',10(I6,4X))
c
          RETURN
          END
```

APPENDIX F

PROGRAM BATCH

```
                            PROGRAM BATCH
c
c       BATCH creates an input file that will be accepted by RPPDM
c
          PARAMETER (MAXNTYPE=7, MAXR=30, MAXS=30)
          INTEGER R, S,MAXRV,MINRV,INCRV,INT,TARGETS
          INTEGER NTAR(MAXNTYPE)
          REAL VTYPE(MAXNTYPE)
c
c       *****************************************************************
c           R        THE MAXIMUM NUMBER OF RV'S AT A SINGLE TARGET
c           S        THE MAXIMUM NUMBER OF INTERCEPTORS AT A SINGLE
c                        TARGET
c         MAXRV      THE MAXIMUM ATTACK SIZE
c         MINRV      THE MINIMUM ATTACK SIZE
c         INCRV      THE ATTACK SIZE INCREMENT
c          INT       THE NUMBER OF INTERCEPTORS
c        TARGETS     THE NUMBER OF TARGETS
c         VTYPE      THE RELATIVE VALUES OF THE TARGET TYPES
c          NTAR      THE NUMBER OF TARGETS FOR EACH TYPE
c       *****************************************************************
c
          INTEGER NROBUST, MAXRRV, MINRRV
c
c       *****************************************************************
c         NROBUST    THE NUMBER OF ROBUST DEFENSES DESIRED
c         MAXRRV     THE MAXIMUM ATTACK SIZE IN THE ROBUST RANGE
c         MINRRV     THE MINIMUM ATTACK SIZE IN THE ROBUST RANGE
c       *****************************************************************
c
c
          INTEGER TER,OUT
          CHARACTER*12 FILEOUT, NAME, ANSWER*1

c       *****************************************************************
c          TER       THE I/O UNIT NUMBER FOR THE TERMINAL
c          OUT       THE I/O UNIT NUMBER FOR THE OUTPUT DEVICE
c          NAME      A CHARACTER VARIABLE SENT BY THE USER
c         ANSWER     A CHARACTER VARIABLE SENT BY THE USER
c         FILEOUT    NAME OF THE OUTPUT DEVICE
c       *****************************************************************
c
c       Set TER to default I/O unit number for terminal
c
          TER=5
c
c       Set OUT to TER+1
c
          OUT=TER+1
c
          WRITE(TER,'(A)') 'OPlease type in the desired file name (of le
     lss than 10 characters'
          WRITE(TER,'(A)') '   including the extension) for storage of t
     lhe parameters ?'
```

```
          READ(TER,'(A)') FILEOUT
          OPEN(UNIT-OUT, FILE-FILEOUT, STATUS-'NEW')
C
C
C       Select output option
C
          WRITE(TER,'(A)') '1You have three options for the output of th
     1e results (with RPPDM): '
          WRITE(TER,'(A)') '0      1) TERMINAL only'
          WRITE(TER,'(A)') '       2) FILE only'
          WRITE(TER,'(A)') '       3) TERMINAL and FILE'
          WRITE(TER,'(A)') 'OPlease enter the number for the desired opt
     1ion ?'
          READ(TER, FMT-*) NOUT
          WRITE(OUT,FMT-*) NOUT
C
C       Ask for file name, if necessary
C
          IF (NOUT .NE. 1) THEN
              WRITE(TER,'(A)') 'OPlease type in the desired file name (o
     1f less than 10 characters'
              WRITE(TER,'(A)') '   including the extension) ?'
              READ(TER,'(A)') NAME
              WRITE(OUT,FMT-'(A12)') NAME
          ENDIF
C
C       Input R and S
C
          WRITE(UNIT-TER, FMT-'(A,I2,A)') 'OThe MAXIMUM number of RV''s
     1(up to ',MAXR,') at a single target ? '
          READ(UNIT-TER, FMT-*) R
          WRITE(UNIT-OUT, FMT-*) R
          WRITE(UNIT-TER, FMT-'(A,I2,A)') 'OThe MAXIMUM number of INTERC
     1EPTORS (up to ',MAXS,') at a single target ? '
          READ(UNIT-TER, FMT-*) S
          WRITE(UNIT-OUT, FMT-*) S
C
C       Select attack methodology
C
          WRITE(UNIT-TER, FMT-'(A)') 'OSelect one of the following attac
     1k methodologies: '
          WRITE(UNIT-TER, FMT-'(A)') '0      1) SIMULTANEOUS ATTACK'
          WRITE(UNIT-TER, FMT-'(A)') '       2) SEQUENTIAL ATTACK'
          WRITE(UNIT-TER, FMT-'(A)') 'OPlease input the number of the de
     1sired attack ? '
          READ(UNIT-TER, FMT-*) NATTYPE
          WRITE(UNIT-OUT, FMT-*) NATTYPE
C
C       Input the failure rate for the RV''s
C
          WRITE(UNIT-TER, FMT-'(A)') 'OThe FAILURE rate of the RV''s ? '
          READ(UNIT-TER, FMT-*) PFA
          WRITE(UNIT-OUT, FMT-*) PFA
C
```

```
c     Select defense methodology
c
          WRITE(UNIT-TER, FMT-'(A)') 'OSelect one of the following defen
     lse methodologies: '
          WRITE(UNIT-TER, FMT-'(A)') 'O     1) ONE SHOT'
          WRITE(UNIT-TER, FMT-'(A)') '      2) SHOOT LOOK SHOOT'
          WRITE(UNIT-TER, FMT-'(A)') 'OPlease input the number for the d
     lesired option ? '
          READ(UNIT-TER, FMT-*) NDFTYPE
          WRITE(UNIT-OUT, FMT-*) NDFTYPE
c
c     If the the attack is sequential, find out whether the defender knows,
c     after the attack begins, the number of RV's slated for each target.
c
          IF (NATTYPE .EQ. 2) THEN
              WRITE(UNIT-TER, FMT-'(A)') 'OIs the defender aware, after
     lthe attack begins, of the number'
              WRITE(UNIT-TER, FMT-'(A)') ' of RV''s slated for each targ
     let (Y or N) ?'
              READ(UNIT-TER, FMT-'(A)') ANSWER
              WRITE(UNIT-OUT, FMT-'(A1)') ANSWER
              IF (ANSWER .EQ. 'Y') THEN
                  IF (NDFTYPE .EQ. 1) THEN
                      WRITE(UNIT-TER, FMT-'(A)')'ONOTE: This scenario is
     1 equivalent to one with a simultaneous attack.'
                  ENDIF
              ENDIF
          ENDIF
c
c       Find the failure rates for the interceptors
c
          IF (NDFTYPE .EQ. 1) THEN
              WRITE(UNIT-TER, FMT-'(A)') 'OThe FAILURE rate of the inter
     lceptors ? '
              READ(UNIT-TER, FMT-*) PFD
          WRITE(UNIT-OUT, FMT-*) PFD
          ELSE
              WRITE(UNIT-TER, FMT-'(A)') 'OThe FAILURE rate for the firs
     lt salvo interceptors ? '
              READ(UNIT-TER, FMT-*) PFD1
              WRITE(UNIT-OUT, FMT-*) PFD1
              WRITE(UNIT-TER, FMT-'(A)') 'OThe FAILURE rate for the seco
     lnd salvo interceptors ? '
              READ(UNIT-TER, FMT-*) PFD2
              WRITE(UNIT-OUT, FMT-*) PFD2
          ENDIF
c
c       Specific parameters of the game
c
          WRITE(TER,'(A)') 'OThe MINIMUM attack size ?'
          READ(TER, FMT-*) MINRV
          WRITE(UNIT-OUT, FMT-*) MINRV
          WRITE(TER,'(A)') 'OThe MAXIMUM attack size ?'
          READ(TER, FMT-*) MAXRV
          WRITE(UNIT-OUT, FMT-*) MAXRV
          WRITE(TER,'(A)') '0The attack size INCREMENT ?'
          READ(TER, FMT-*) INCRV
          WRITE(UNIT-OUT, FMT-*) INCRV
          WRITE(TER,'(A)') '0The NUMBER of interceptors ?'
          READ(TER, fmt-*) INT
          WRITE(UNIT-OUT, FMT-*) INT
          WRITE(TER,'(A)') '0The TOTAL NUMBER of targets ?'
          READ(TER, FMT-*) TARGETS
          WRITE(UNIT-OUT, FMT-*) TARGETS
          WRITE(TER,'(A)') '0The number of TYPES of targets ?'
          READ(TER, FMT-*) NTYPE
          WRITE(UNIT-OUT, FMT-*) NTYPE
c
          IF (NTYPE .NE. 1) THEN
              WRITE(TER, '(A)') '0Enter first the RELATIVE VALUE and the
     1n the NUMBER of targets'
              WRITE(TER, '(A)') ' for each type. Separate the two entrie
     1s for each target type with'
              WRITE(TER, '(A)') ' a comma and hit <CR> following the ent
     1ries for each target type: '
c
c           Loop through each target type
c
              DO 100 I - 1, NTYPE
                  READ(TER, fmt-*) VTYPE(I), NTAR(I)
                  WRITE(UNIT-OUT, FMT-*) VTYPE(I), ',', NTAR(I)
                  TNT-TNT+NTAR(I)
100           CONTINUE
              IF (TNT .NE. TARGETS) THEN
                  WRITE(TER, '(A)') '0The sum of the targets in the indi
     1vidual target types does not'
                  WRITE(TER, '(A)') ' equal the total number of targets'
                  STOP
              ENDIF
          ENDIF
c
c
          WRITE(TER,'(A)') '1Please enter the number of different ranges
     1 of RV''s for which robust '
          WRITE(TER,'(A)') '   solutions are to be found ?'
          WRITE(TER,'(A)') '          ***** Enter 0 if no robust solution
     1 is desired*****         '
          READ(TER,*) NROBUST
          WRITE(UNIT-OUT, FMT-*) NROBUST
          IF (NROBUST .EQ. 0) THEN
              STOP
          ENDIF
          WRITE(TER,'(A)') '0The lower and upper bounds for the RV range
     1s must be between'
          WRITE(TER,'(I5,'' and '',I5)') MINRV, MAXRV
c
          DO 200 I1-1, NROBUST
              WRITE (TER,'(A)') '0     The lower bound :'
              READ (TER,*) MINRRV
              WRITE(UNIT=OUT, FMT=*) MINRRV
              WRITE (TER,'(A)') '0     The upper bound :'
              READ (TER,*) MAXRRV
              WRITE(UNIT=OUT, FMT=*) MAXRRV
              WRITE(TER,'(//)')
              IF (I1 .NE. 1) THEN
                  WRITE(TER,'(A)') ' NEXT'
              ENDIF
              WRITE(TER,'(//)')
200       CONTINUE
c
          STOP
          END
```

APPENDIX G

DERIVATION OF THE LP'S USED IN BG AND YROBUST

1. Basic Game

Let $\overline{K}$ be the set of all target types,
T be the total number of targets,
RV be the number of reentry vehicles attacking the targets, and
INT be the defending interceptors.

Let R be the maximum number of RVs allowed to attack a single target and
S be the maximum number of interceptors allowed to defend a single target.

Let $VF_k = V_k \bullet N_k / \sum_{k \in \overline{K}} (V_k \bullet N_k)$

and $NF_k = N_k / \sum_{k \in \overline{K}} N_k$

where V_k = the relative value of a type k target,

N_k = the number of targets of type k,

and k is a target type in $\overline{K}$.

Define a strategy X (or Y) as the set of X^k (or Y^k) for all $k \in \overline{K}$ where

$$X^k = (X_0^k, \ldots, X_R^k)$$
$$Y^k = (Y_0^k, \ldots, Y_S^k)$$

and

X_i^k = fraction of type k targets assigned i RVs

Y_j^k = fraction of type k targets assigned j interceptors.

Let P be the matrix $[P_{ij}]$

and P_i the ith row of P,

where an element P_{ij} is the probability that a target under attack by i RVs and defended by j interceptors will survive.

The formation of the minimax problem is as follows.

Let VBG = expected fraction of the total value $\sum_{k \in \overline{K}} (N_k \bullet V_k)$ that will survive.

$$VBG = \max_{Y^k \geq 0} \quad \min_{X^k \geq 0} \quad \sum_{k \in \overline{K}} (VF_k \bullet X^{k''} \bullet P \bullet Y^k)$$

$$\sum_{j=0}^{S} Y_j^k = 1, k \in \overline{K} \qquad \sum_{i=0}^{R} X_i^k = 1, k \in \overline{K}$$

$$\sum_{k \in \overline{K}} \sum_{j=0}^{S} (j \bullet Y_j^k \bullet NF_k) = \frac{INT}{T} \quad \sum_{k \in \overline{K}} \sum_{i=0} i \bullet X_i^k \bullet NF_k = \frac{RV}{T},$$

where $X^{k''}$ is X^k transposed.

Taking the dual of the inside problem yields the following linear program.

LP1 is:

$$VBG = \max_{Y_j^k \geq 0, s_k, t} \left[\sum_{k \in \overline{K}} s_k - \frac{RV}{T} \cdot t \right]$$

subject to

$$\left.\begin{array}{ll} s_k & \leq VF_k \cdot P_0 \cdot Y^k \\ s_k - NF_k \cdot t & \leq VF_k \cdot P_1 \cdot Y^k \\ & \bullet \\ & \bullet \\ & \bullet \\ s_k - R \cdot NF_k \cdot t & \leq VF_k \cdot P_R \cdot Y^k \\ \sum_{j=0}^{S} Y_j^k = 1 & \end{array}\right\} \text{ for all } k \in \overline{K}$$

$$\sum_{k \varepsilon \overline{K}} \sum_{j=0}^{S} j \cdot Y_j^k \cdot NF_k = INT/T,$$

$$Y_j^k \geq 0, \text{ and } s_k \text{ and t unrestricted.}$$

The Y which yields VBG is the optimal minimax defender's strategy, Y*.

X*, the attacker's minimax strategy, is equivalent to the dual variables of the inequality constraints.

In addition s_k and t may be assumed to be positive. (See [1] on Page R-1.)

2. Robust Game

Assume that the attacker can always discover and thus optimize against Y. The defender wishes to find the robust strategy YII which will solve the following problem:

$$\max_{Y \in Y_p} \min_{A \in \overline{A}} \{R_A(X,Y)\}$$

where:

(1) Y_p is the set of all Y's such that

$$\sum_{j=0}^{S} Y_j^k = 1 \text{ for all } k \in \overline{K}$$

$$\sum_{k \in \overline{K}} \sum_{j=0} \left(j \cdot NF_k \cdot Y_j^k\right) = \frac{INT}{T},$$

(2) $\bar{A}$ is the set of possible attack sizes, and

$$(3)\ R_A(X,Y) = \frac{1}{VBG(A)} \min_{\text{the set of all } X^k = (X_0^k,\ldots,X_R^k)} \sum_{k \in \bar{K}} (VF_k \bullet X^{k''} \bullet P \bullet Y^k)$$

where

$$\sum_{i=0}^{R} X_i^k = 1 \text{ for all } k \in \bar{K} \text{ and}$$

$$\sum_{k \in \bar{K}} \sum_{i=0}^{R} (i \bullet NF_k \bullet X_i^k) = \frac{RV(A)}{T}.$$

Taking the dual,

$$R_A(X,Y) = \frac{1}{VBG(A)} \max_{Y_j^k \geq 0,\ s_k(A),\ t(A)} \left[\sum_{k \in \bar{K}} s_k(A) - \frac{RV(A)}{T} t(A) \right]$$

subject to

$$\left.\begin{array}{l} s_k(A) \leq VF_k \bullet P_0 \bullet Y_k \\ s_k(A) - R \bullet NF_k \bullet t(A) \leq VF_k \bullet P_1 \bullet Y^k \\ \vdots \\ s_k(A) - R \bullet NF_k \bullet t(A) \leq VF_k \bullet P_R \bullet Y^k \end{array}\right\} \text{for all } k \in \bar{K}.$$

Inserting this back into the original problem we have:

$$\max_{Y \in Y_p} \min_{A \in \bar{A}} \left[\frac{1}{VBG(A)} \max_{s_k(A), t(A)} \sum_{k \in \bar{K}} s_k(A) - \frac{RV(A)}{T} t(A) \right]$$

which yields the following linear program.

LP2 is:

$$\max_{Y_j^k \geq 0,\ s_k(A),\ t(A)} \rho$$

subject to

$$\left.\begin{array}{l} VBG(A)\ \ \rho \leq \sum_{k \in \bar{K}} s_k(A) - \frac{RV(A)}{T}\, t(A) \\ \left.\begin{array}{ll} s_k(A) & \leq VF_k \bullet P_0 \bullet Y^k \\ s_k(A) - NF \bullet t(A) & \leq VF_k \bullet P_1 \bullet Y^k \\ & \vdots \\ s_k(A) - R \bullet NF_k \bullet t(A) & \leq VF_k \bullet P_R \bullet Y^k \end{array}\right\} k \in \bar{K} \end{array}\right\} A \in \bar{A}.$$

Once more we can assume $s_k(A)$ and $t(A)$ to be nonnegative (see [1] on Page R-1).

(Substituting $Z_k(i)$ for $VF_k \bullet P_i \bullet Y^k$ yields the LP used in YROBUST.)

By assumption, the attacker "knows" YII, the Y which solves LP2. Thus we can find XII, the optimal attack against YII, and VII, the expected survival rate for any attack size A, by solving the following linear program:

LP3 is:

$$VII(A) = \min_{X^k \geq 0} \sum_{k \in \bar{K}} (VF_k \bullet X^{k''} \bullet P \bullet YII^k)$$

subject to
$$\sum_{i=0}^{R} X_i^k = 1 \text{ for all } k \in \bar{K},$$

$$\sum_{k \in \bar{K}} \sum_{i=0}^{R} i \bullet X_i^k \bullet NF_k = \frac{RV(A)}{T}.$$

APPENDIX H

EQUATIONS USED IN SIMAT1, SIMAT2, SEQAT1, AND SEQAT2

TABLE OF CONTENTS

1. Equation for SIMAT1--Simultaneous Attack with One Opportunity to Shoot[1]

We assume that a single target is under a simultaneous attack by A identical missiles, and is being defended by D identical interceptors. Let

d = probability that a defending interceptor will destroy the attacking at which it is directed,

a = probability that an attacking missile will destroy the target, given that it evades all defending interceptors.

We assume that the defense can see the entire attack, and must decide on the number of interceptors that it assigns to each of the attacking missiles.

Given suitable independence assumptions, the probability of the target surviving an attack of n_j attacking missiles, each of which are being attacked by j defending interceptors, is

$$(1 - a\,(1 - d)^j)^{n_j}\,. \qquad (1)$$

Thus, if the target is attacked by A missiles and, for each j between 0 and D, the defender assigns a total of jn_j interceptors against n_j missiles, such that j interceptors are assigned against each of these n_j missiles, where

$$\sum_{j=0}^{D} n_j = A$$

and

$$\sum_{j=0}^{D} j n_j = D,$$

then the probability that the target survives is

$$\prod_{j=0}^{D} (1 - a\,(1 - d)^j)^{n_j}\,. \qquad (2)$$

[1] Source: Appendix A of Reference [1] on page R-1.

The defender wishes to select the n_j in order to maximize this probability.

We wish to show that the "uniform defense" obtained by spreading the D interceptors as equally as possible over the A attackers is optimal.

Consider an allocation of interceptors to attackers which is not uniform. Then there is a pair $i < j$ with $n_i, n_j > 0$ where $i + 2 \leq j$. Consider a new (and more uniform) allocation obtained by allocating $i + 1$ interceptors to one of the n_i attackers, and $j - 1$ interceptors to one of the n_j attackers. The probability that the target now survives is

$$\frac{(1 - a\,(1 - d)^{i+1})\,(1 - a\,(1 - d)^{j-1})}{(1 - a\,(1 - d)^{i})\,(1 - a\,(1 - d)^{j})} \qquad .$$

times the old probability, and this is easily shown to be greater than 1.

The most uniform of defenses assigns

$i = [D/A]$ defenders to $n_i = A\,(1 - \langle D/A \rangle)$ attackers

and

$j = [D/A] + 1$ defenders to $n_j = A \langle D/A \rangle$ attackers

(where [x] and <x> denote the integer and fractional parts of x). Thus, the optimal defense is:

$$n_j = \begin{cases} A\,(1 - \langle D/A \rangle) & \text{for } j = [D/A] \\ A \langle D/A \rangle & \text{for } j = [D/A] + 1 \\ 0 & \text{otherwise.} \end{cases}$$

Substituting these values for n_j into (2) yields

$$P(A,D) = (1-a(1-d)^{[D/A]+1})^{A\langle D/A \rangle}(1-a(1-d)^{[D/A]})^{A(1-\langle D/A \rangle)},$$

where P(A,D) is the probability the target survives given that it is attacked by A missiles and defended by D interceptors that are allocated against these attacking missiles according to this uniform defense. Table 1 gives the numerical values of P(A,D) for four examples.

Table 1. VALUES OF P(A,D) (SIMULTANEOUS ATTACK)

a = .7 d = .7

A \ D	0	1	2	3	4	5	6	7	8	9	10
0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1	.3000	.7900	.9370	.9811	.9943	.9983	.9995	.9998	1.0000	1.0000	1.0000
2	.0900	.2370	.6241	.7402	.8780	.9193	.9626	.9755	.9887	.9926	.9966
3	.0270	.0711	.1872	.4930	.5848	.6936	.8227	.8614	.9019	.9444	.9571
4	.0081	.0213	.0562	.1479	.3895	.4620	.5479	.6499	.7708	.8071	.8451
5	.0024	.0064	.0169	.0444	.1169	.3077	.3650	.4329	.5134	.6090	.7223
6	.0007	.0019	.0051	.0133	.0351	.0923	.2431	.2883	.3420	.4056	.4811
7	.0002	.0006	.0015	.0040	.0105	.0277	.0729	.1920	.2278	.2702	.3204
8	.0001	.0002	.0005	.0012	.0032	.0083	.0219	.0576	.1517	.1799	.2134
9	.0000	.0001	.0001	.0004	.0009	.0025	.0066	.0173	.0455	.1199	.1422
10	.0000	.0000	.0000	.0001	.0003	.0007	.0020	.0052	.0137	.0360	.0947

T-10-88-1

a = .7, d = .9

A \ D	0	1	2	3	4	5	6	7	8	9	10
0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1	.3000	.9300	.9930	.9993	.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	.0900	.2790	.8649	.9235	.9860	.9923	.9986	.9992	.9999	.9999	1.0000
3	.0270	.0837	.2595	.8044	.8588	.9170	.9791	.9854	.9916	.9979	.9985
4	.0081	.0251	.0778	.2413	.7481	.7987	.8528	.9106	.9723	.9785	.9847
5	.0024	.0075	.0234	.0724	.2244	.6957	.7428	.7931	.8469	.9042	.9655
6	.0007	.0023	.0070	.0217	.0673	.2087	.6470	.6908	.7376	.7876	.8409
7	.0002	.0007	.0021	.0065	.0202	.0626	.1941	.6017	.6425	.6860	.7325
8	.0001	.0002	.0006	.0020	.0061	.0188	.0582	.1805	.5596	.5975	.6380
9	.0000	.0001	.0002	.0006	.0018	.0056	.0175	.0542	.1679	.5204	.5557
10	.0000	.0000	.0001	.0002	.0005	.0017	.0052	.0162	.0504	.1561	.4840

T-10-88-2

a = .9, d = .7

A \ D	0	1	2	3	4	5	6	7	8	9	10
0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1	.1000	.7300	.9190	.9757	.9927	.9978	.9993	.9998	.9999	1.0000	1.0000
2	.0100	.0730	.5329	.6709	.8446	.8967	.9520	.9686	.9855	.9905	.9956
3	.0010	.0073	.0533	.3890	.4897	.6165	.7762	.8240	.8749	.9289	.9451
4	.0001	.0007	.0053	.0389	.2840	.3575	.4501	.5666	.7133	.7573	.8040
5	.0000	.0001	.0005	.0039	.0284	.2073	.2610	.3285	.4136	.5207	.6555
6	.0000	.0000	.0001	.0004	.0028	.0207	.1513	.1905	.2398	.3019	.3801
7	.0000	.0000	.0000	.0000	.0003	.0021	.0151	.1105	.1391	.1751	.2204
8	.0000	.0000	.0000	.0000	.0000	.0002	.0015	.0110	.0806	.1015	.1278
9	.0000	.0000	.0000	.0000	.0000	.0000	.0002	.0011	.0081	.0589	.0741
10	.0000	.0000	.0000	.0000	.0000	.0000	.0000	.0001	.0008	.0059	.0430

T-10-88-3

a = .9, d = .9

A \ D	0	1	2	3	4	5	6	7	8	9	10
0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1	.1000	.9100	.9910	.9991	.9999	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
2	.0100	.0910	.8281	.9018	.9821	.9901	.9982	.9990	.9998	.9999	1.0000
3	.0010	.0091	.0828	.7536	.8206	.8937	.9732	.9812	.9892	.9973	.9981
4	.0001	.0009	.0083	.0754	.6857	.7468	.8133	.8857	.9645	.9724	.9803
5	.0000	.0001	.0008	.0075	.0686	.6240	.6796	.7401	.8059	.8777	.9558
6	.0000	.0000	.0001	.0008	.0069	.0624	.5679	.6184	.6735	.7334	.7987
7	.0000	.0000	.0000	.0001	.0007	.0062	.0568	.5168	.5628	.6129	.6674
8	.0000	.0000	.0000	.0000	.0001	.0006	.0057	.0517	.4703	.5121	.5577
9	.0000	.0000	.0000	.0000	.0000	.0001	.0006	.0052	.0470	.4279	.4660
10	.0000	.0000	.0000	.0000	.0000	.0000	.0001	.0005	.0047	.0428	.3894

T-10-88-4

2. Equation for SIMAT2--Simultaneous Attack with Two Opportunities to Shoot

The following material is extracted from Richard M. Soland, "Optimal Terminal Defense Tactics Against Simultaneously Arriving RVs When Several Sequential Engagements are Possible," TR-85/12, Institute for Reliability and Risk Analysis, School of Engineering and Applied Science, The George Washington University, 18 October 1985. References for this section are given on page H-18.

a. INTRODUCTION

An important and frequent element of ballistic missile defense models is the presence and role of a terminal defense that defends a single target, either an area target such as a city or a point target such as an ICBM silo. Given that such a terminal defense consists of a given number of interceptors, the question immediately arises as to the tactics to be employed by the defense in using those interceptors and how the expected damage to the target varies as a function of the number of attacking reentry vehicles (RVs).

The RVs may be assumed to arrive sequentially, so that the defense never knows how many are coming, or else simultaneously, so that full knowledge of the attack size is obtained. Intermediate cases may be more realistic of course. The defense may have morethan one opportunity to engage some or all of the RVs, and it may receive information on the results of some engagements before undertaking subsequent engagements. Here we treat the case in which RVs are assumed to arrive simultaneously and a fixed number of engagements of each RV are possible, with shoot-look-shoot capability between them.

The scenario we consider is as follows: a single target is attacked simultaneously by A reentry vehicles; it is defended by a terminal defense consisting of D interceptors. The RVs act independently, and the expected fraction of the target destroyed by each unintercepted RV is ρ, where $0 < \rho \leq 1$. The defense may engage each RV up to K times, with one or more interceptors used at each engagement, and it has shoot-look-shoot capability between successive engagements. That is, the defense observes whether or not a particular RV survives a particular engagement before deciding how many interceptors, if any, to use against it at the next engagement. All interceptors are assumed to operate independently, even when used simultaneously against the same RV. The single-shot kill probability of one interceptor against one RV may vary with the number of the engagement in the sequence of K engagements.

The objective of the defense is to minimize the expected fraction of the target destroyed. A policy for the defense is a rule that specifies for each engagement, how many interceptors to use against each of the RVs that has not been destroyed at previous engagements. We desire to find an optimal policy as a function of A, D, and K.

Chapter 3 of Eckler and Burr (1972) also deals with the tactics to be employed by a terminal defense and presents several different models, each based on specific assumptions. One model, that of section 3.5.1, is a special case of ours; it corresponds to $K = 2$ and $\rho = 1$. Some typical results are presented, but details of the methodology are omitted.

Burr et al. (1985) and Falk (1985b) treat the case in which RVs arrive sequentially but only one engagement of each RV is possible, and Falk (1985a) extends such analyses to the case of several engagements and shoot-look-shoot capability between them. In these three papers the optimality criterion is minimization of the number of interceptors D required to guarantee that the expected fraction of the target destroyed, as a function of A, lies below a given bounding function. Our criterion is minimization of the expected fraction of the target destroyed; as we show, however, with a little additional work we can also treat this other criterion.

A brief outline of the paper is as follows. In the next section we define notation and point out the intuitive result (which is proved in the Appendix) that at each engagement the interceptors used should be spread as uniformly as possible among the attacking RVs. Using this result, we present a dynamic programming algorithm that determines optimal policies. The following section presents several extensions of the basic model; two of them deal, respectively, with the expected number of interceptors remaining after the attack and determination of the minimum number of interceptors needed to provide a desired level of protection of the target. The final section contains illustrative numerical examples.

b. ANALYSIS

It is convenient to number the engagements in reverse chronological order, so we shall refer generally to there being k engagements remaining before the end of the attack. Here $k = 0,1,\ldots,K$, where $k = 0$ indicates that no further engagements are possible. For $k=1,\ldots,K$, let p_k be the single-shot kill probability of one interceptor against one RV on the kth engagement before the end, and let $q_k = 1 - p_k$. We assume that each q_k satisfies $0 < q_k < 1$; the contrary cases are not of interest.

We define

S(a,d,k) = the expected fraction of the target destroyed if k engagements remain, the defense has d interceptors left, there are a RVs left undestroyed, and the defense follows an optimal policy for the remaining k engagements.

S(a,d,k) is defined for a=0,1,...,A; d=0,1,...,D; k=0,1,...,K. Now define, for j=0,1,...,a; a=0,1,...,A; i=0,1,...,d; d=0,1,...,D; k-1,...,K,

P(j|a,i,d,k) = the probability that j RVs survive the kth engagement from the end when there are a RVs left before that engagement and i of the d remaining interceptors are used in an optimal manner at that engagement.

The principle of optimality of dynamic programming now allows us to write the following recursive equation which is valid for a=1,...,A; d=0,1,...,D; k=1,...,K:

$$S(a,d,k) = \min_{i=0,\ldots,d} \left\{ \sum_{j=0}^{a} P(j|a,i,d,k)\, S(j,d-i,k-1) \right\}. \qquad (1)$$

In order to use this recursion to calculate S(a,d,k), we need appropriate boundary conditions in the form of S(0,d,k) and S(a,d,0). These are clearly as follows:

$S(0,d,k) = 0$ for $d=0,1,\ldots,D$; $k=1,\ldots,K$,

$S(a,d,0) = 1 - (1-p)^a$ for $a=0,1,\ldots,A$; $d=0,1,\ldots,D$.

It remains to provide the $P(j|a,i,d,k)$ before the above recursion can be used. This is rendered relatively straightforward by the following result, whose proof is provided in the Appendix: an optimal manner in which to use i of d interceptors against a RVs is to spread them as uniformly as possible among the a RVs. Thus if i/a is an integer, say I, each of the a RVs is assigned I interceptors and has survival probability q_k^I. The number of RVs that survive the engagement thus has a binomial distribution, so

$$P(j|a,i,d,k) = \binom{a}{j} q_k^{jI} (1-q_k^I)^{a-j} .$$

More generally, i/a is not an integer. Then it is easily shown that $a + a\lfloor i/a \rfloor - i$ of the a RVs are assigned $\lfloor i/a \rfloor$ interceptors each and the remaining $i - a\lfloor i/a \rfloor$ RVs are assigned $\lceil i/a \rceil$ interceptors each (here $\lfloor x \rfloor$ is the greatest integer less than or equal to x and $\lceil x \rceil$ is the smallest integer greater than or equal to x). From this it follows that the number of RVs that survive the engagement is the sum of two independent binomial random variables with slightly different success probabilities. The needed probabilities $P(j|a,i,d,k)$ may then be obtained by numerically convoluting the two binomial distributions in question.

An alternative way to obtain the needed probabilities, the one actually used for our computational results, is as follows. Let J be the random variable who probability distribution is $P(j|a,i,d,k)$.

Then J is the sum of a independent Bernoulli random variables, each being equal to one if a specific one of the a RVs survives the engagement, and equal to zero otherwise. We may thus write $J = \Sigma_{\ell=1}^{a} X_\ell$, where $P(X_\ell = 1) = r_\ell$ and $P(X_\ell = 0) = 1 - r_\ell$. The r_ℓ are given by $r_\ell = q_k^\alpha$ for $\ell=1,\ldots,a + a\lfloor i/a \rfloor - i$ and $r_\ell = q_k^\beta$ for $\ell = a + a\lfloor i/a \rfloor - i + 1,\ldots,a$, where $\alpha = \lfloor i/a \rfloor$ and $\beta = \lceil i/a \rceil$. By defining the random variables $J_s \equiv \Sigma_{\ell=1}^{s} X_\ell$ for $s = 1,\ldots,a$, so that $J = J_a$, we can calculate the probability distributions of the J_s successively from the recursion (which is valid for $j=1,\ldots,s$ and $s=2,\ldots,a$)

$$P(J_s = j) = (1-r_s)\, P(J_{s-1}=j) + r_s P(J_{s-1}=j-1)$$

along with the boundary conditions

$$P(J_s = 0) = \prod_{\ell=1}^{s} (1-r_\ell), \qquad P(J_1=1) = r_1.$$

A closed-form expression for S(a,d,1) is sometimes useful, and it is easily obtained from the uniform-defense property and fairly simple probabilistic analysis; the result is

$$S(a,d,1) = 1 - (1 - \rho q_1^\alpha)^{a+a\alpha-d}\,(1-\rho q_1^\beta)^{d-a\alpha},$$

where $\alpha = \lfloor d/a \rfloor$ and $\beta = \lceil d/a \rceil$.

C. EXTENSIONS OF THE BASIC MODEL

In anticipation of the possibility of another attack after the present one is over, the terminal defense may be interested in, as a secondary criterion, the expected number of interceptors remaining to it at the end of the current attack. We thus define

$T(a,d,k)$ = the expected number of interceptors left after the attack if k engagements remain, the defense presently has d interceptors left, there are a RVs left undestroyed, and the defense follows an optimal policy for the remaining k engagements.

Like $S(a,d,k)$, $T(a,d,k)$ is defined for $a=0,1,\ldots,A$; $d=0,1,\ldots,D$; $k=0,1,\ldots,K$. We can calculate $T(a,d,k)$ along with $S(a,d,k)$ as follows: Let i^* be a minimizing value of i in the recursion (1) used to determine $S(a,d,k)$. Then

$$T(a,d,k) = \sum_{j=0}^{a} P(j\,|\,a,i^*,d,k)\; T(j,d-i^*,k-1), \qquad (2)$$

for $a=1,\ldots,A$; $d=0,1,\ldots,D$; $k=1,\ldots,K$. The necessary boundary conditions are

$$T(0,d,k) = d \quad \text{for} \quad d=0,1,\ldots,D;\ k=0,1,\ldots,K,$$

$$T(a,d,0) = d \quad \text{for} \quad a=1,\ldots,A;\ d=0,1,\ldots,D.$$

The recursion (2) does not serve to uniquely determine the $T(a,d,k)$ unless the minimizing i^* in (1) are unique. But the index i^* is not necessarily unique, so we adopt the following convention, which then serves to determine a unique value for $T(a,d,k)$. If i^* in (1) is not unique, use the smallest value of i^* that maximizes $T(a,d,k)$ as computed by (2).

The boundary condition $S(a,d,0) = 1 - (1-\rho)^a$ given above was based on the assumption of RVs that act independently and, if unintercepted, each destroy an expected fraction ρ of the target. The dynamic programming scheme given works equally well with the more general boundary condition $S(a,d,0) = g(a)$ for $a=0,1,\ldots,A$; $d=0,1,\ldots,D$, where g is nondecreasing, $g(0) = 0$, and $g(A) \leq 1$. This allows use of an arbitrary relationship between the number of unintercepted RVs and the expected fraction of the target destroyed.

In damage-limitation studies it is sometimes of interest to determine the smallest number of interceptors needed by the defense to provide a desired level of protection of the target. One way to interpret the phrase "desired level of protection" is by specifying a nondecreasing maximum damage function f and requiring that the expected fraction of the target destroyed by a RVs not exceed f(a) for a=1,...,A. For example, see Burr et al. (1985) and Falk (1985b) for analyses of this problem when the RVs are assumed to arrive sequentially and the defense has no shoot-look-shoot capability, and see Falk (1985a) for an extension to the case in which the defense does have shoot-look-shoot capability. In the present context, and for fixed A and K, we may phrase the problem as

$$\begin{aligned} &\text{Minimize} \quad D \\ &\text{subject to} \quad S(a,D,K) \leq f(a), \quad a=1,\ldots,A. \end{aligned}$$

We can solve this problem easily by continuing the computations of the dynamic programming scheme for successively larger values of D until one is found that satisfies all the indicated constraints.

d. EXAMPLES

For the case $\rho = 0.7$, $p_1 = 0.8$ and $p_2 = 0.9$, Table 1 gives $S(A,D,2)$, i^*, and $T(A,D,2)$ for $A,D = 1(1)10$; the three quantities appear in respective rows. For example, with $A = 7$ RVs, $D = 9$ interceptors and 2 engagements left, the defense should use $i^* = 7$ interceptors and hold 2 in reserve for possible use at the last engagement. The expected fraction of the target destroyed is $S(7,9,2) = 0.0631$ and the expected number of interceptors left after the attack is $T(7,9,2) = 0.957$.

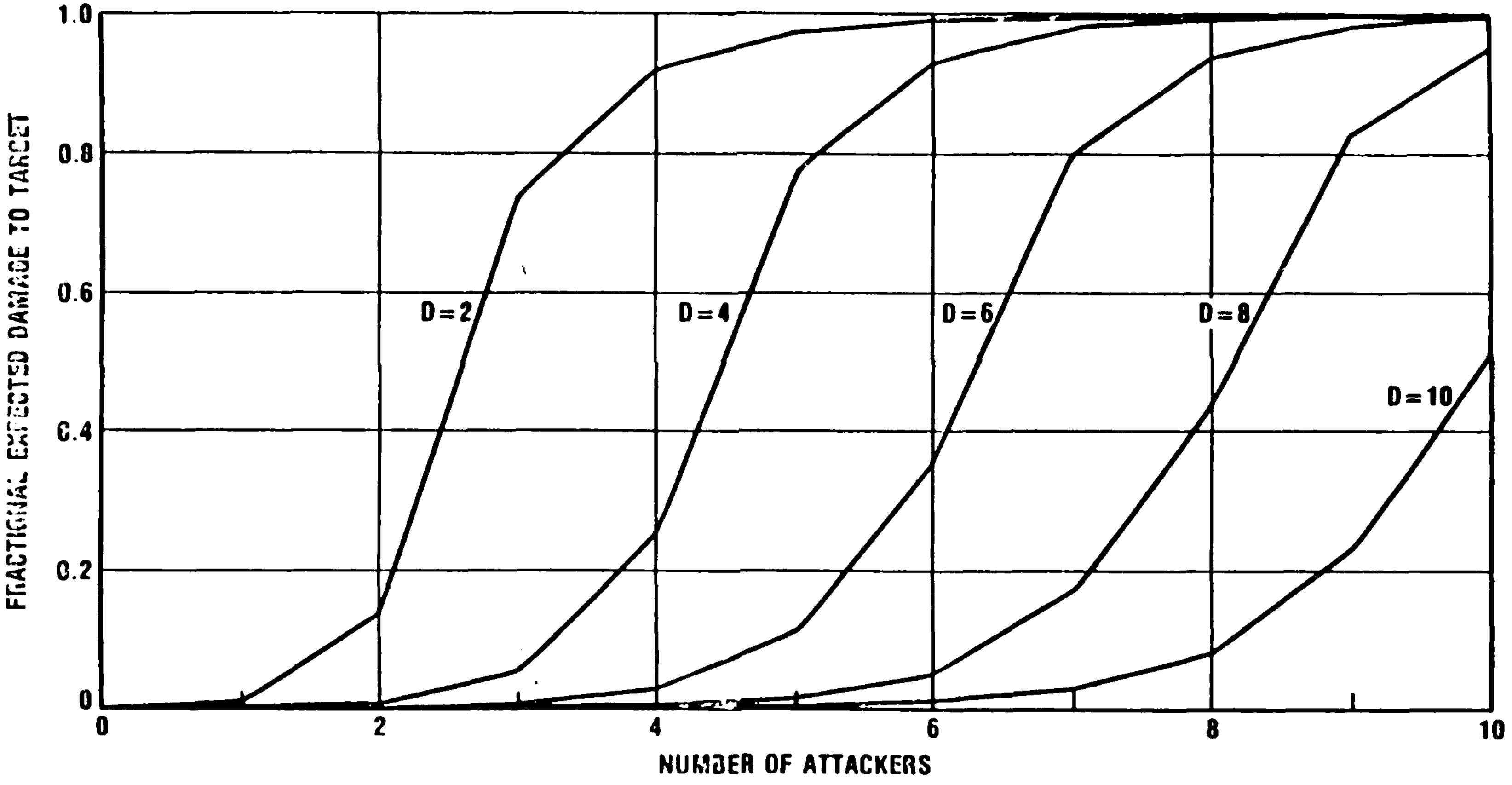
Figure 1 shows the expected fractional damage $S(A,D,2)$ as a function of A for $D = 2(2)10$.

Figure 2 shows solutions of the damage limitation problem minimize D subject to $S(a,D,K) \leq f(a)$, $a=1,...,10$, for the linear function $f(a) = sa$. Solutions are given for a range of values of the slope s and for both $K = 1$ and $K = 2$; the other parameters of the problem are $\rho = 0.9$ and $p_1 = p_2 = 0.7$. For example, if it is desired to have a defense whose expected fractional damage per attacking RV is limited to $s = 0.125$, then 9 interceptors are needed if only one engagement is possible $(K = 1)$, but only 8 interceptors are required if two engagements are possible $(K = 2)$.

TABLE 1

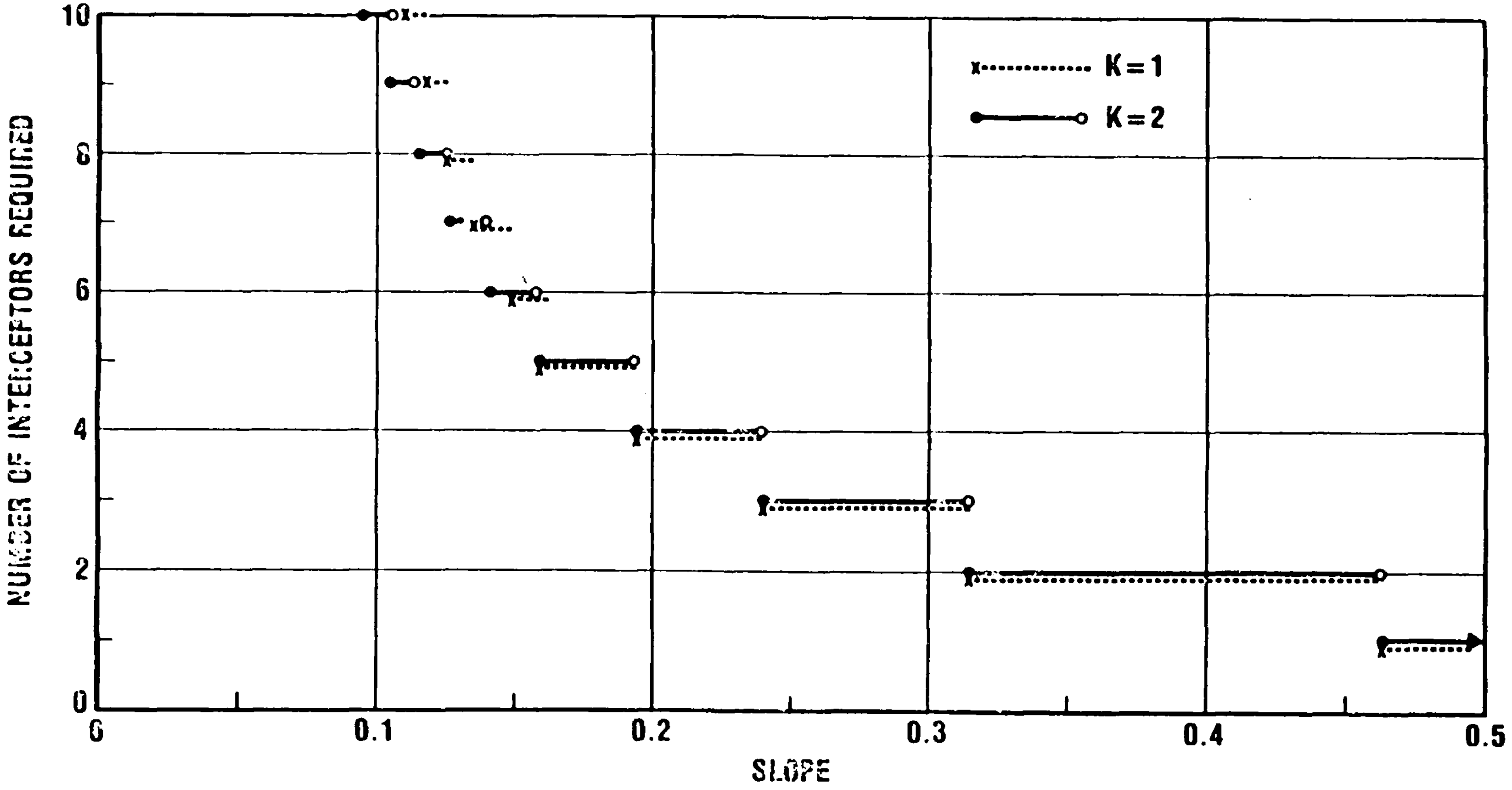
Results for 2 Engagements when $p_1 = 0.8$, $p_2 = 0.9$ and $\rho = 0.7$

```
TABLE OF EXPECTED FRACTION OF TARGET DESTROYED, OPTIMAL NUMBER OF
INTERCEPTORS TO USE, AND EXPECTED NUMBER OF INTERCEPTORS LEFT OVER

          1=A     2=A     3=A     4=A     5=A     6=A     7=A     8=A     9=A    10=A

D= 1 0.0700 0.7210 0.9163 0.9749 0.9925 0.9977 0.9993 0.9998 0.9999 1.0000
D= 1      1      1      1      1      1      1      1      1      1      1
D= 1  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000

D= 2 0.0070 0.1351 0.7405 0.9222 0.9766 0.9930 0.9979 0.9994 0.9998 0.9999
D= 2      2      2      2      2      2      2      2      2      2      2
D= 2  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000

D= 3 0.0007 0.0326 0.1956 0.7587 0.9276 0.9783 0.9935 0.9980 0.9994 0.9998
D= 3      3      2      3      3      3      3      3      3      3      3
D= 3  0.000  0.810  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000

D= 4 0.0001 0.0076 0.0550 0.2519 0.7756 0.9327 0.9798 0.9939 0.9982 0.9995
D= 4      4      2      3      4      4      4      4      4      4      4
D= 4  0.000  1.620  0.729  0.000  0.000  0.000  0.000  0.000  0.000  0.000

D= 5 0.0000 0.0026 0.0146 0.0803 0.3043 0.7913 0.9374 0.9812 0.9944 0.9983
D= 5      5      2      3      4      5      5      5      5      5      5
D= 5  0.000  2.430  1.458  0.656  0.000  0.000  0.000  0.000  0.000  0.000

D= 6 0.0000 0.0006 0.0062 0.0237 0.1079 0.3530 0.8059 0.9418 0.9825 0.9948
D= 6      6      4      3      4      5      6      6      6      6      6
D= 6  0.000  1.960  2.187  1.312  0.590  0.000  0.000  0.000  0.000  0.000

D= 7 0.0000 0.0001 0.0020 0.0110 0.0349 0.1373 0.3983 0.8195 0.9458 0.9838
D= 7      7      4      3      4      5      6      7      7      7      7
D= 7  0.000  2.940  2.916  1.968  1.181  0.531  0.000  0.000  0.000  0.000

D= 8 0.0000 0.0000 0.0009 0.0041 0.0171 0.0481 0.1679 0.4404 0.8321 0.9496
D= 8      8      4      4      4      5      6      7      8      8      8
D= 8  0.000  3.920  3.208  2.624  1.771  1.063  0.478  0.000  0.000  0.000

D= 9 0.0000 0.0000 0.0002 0.0024 0.0069 0.0245 0.0631 0.1994 0.4796 0.8439
D= 9      9      4      6      4      5      6      7      8      9      9
D= 9  0.000  4.901  2.911  3.280  2.362  1.594  0.957  0.430  0.000  0.000

D=10 0.0000 0.0000 0.0000 0.0009 0.0042 0.0105 0.0330 0.0799 0.2313 0.5160
D=10     10      4      6      4      5      6      7      8      9     10
D=10  0.000  5.881  3.881  3.937  2.952  2.126  1.435  0.861  0.387  0.000
```



10-18-86-17

Figure 1. Expected Damage Curves for Various Defense Levels D



10-18-85-16

Figure 2. Number of Interceptors Required for a Given Slope

ANNEX

Here we provide a verification of the intuitive result that, regardless of the number of engagements remaining, an optimal manner in which to use i of d interceptors against a RVs is to spread them as uniformly as possible among the a RVs, i.e., use a uniform defense.

Suppose there are $k \geq 1$ engagements remaining and suppose there is an optimal way to use i of d interceptors against the a RVs that is not uniform, i.e., there are two RVs such that one of them is assigned at least 2 more interceptors than the other. We will show that the expected damage done to the target is not increased if one interceptor is switched from the heavily attacked RV to the lightly attacked RV. By repetition of this step a finite number of times it then follows that a uniform defense is optimal.

Let $u(\ell)$ be the number of interceptors assigned to RV ℓ $(\ell=1,\ldots,a)$, so that $\Sigma_{\ell}\ u(\ell) = i$. Without loss of generality, we may assume $u(1) \geq u(2) + 2$. Define random variables $X_1,\ldots,X_a$ as $X_{\ell} = 1$ if RV ℓ survives its attack by $u(\ell)$ interceptors and $X_{\ell} = 0$ otherwise.

Let $X_{12} = X_1 + X_2$. Then $J \equiv X_{12} + \Sigma_{\ell=3}^{a}\ X_{\ell}$ is the number of RVs that survive the current engagement and $E[S(J,\ d-i,\ k-1)]$ is the expected fractional damage done to the target.

Now consider the alternative defense defined by $u\ (1) = u(1) -1$, $u'(2) = u(2) + 1$, and $u'(\ell) = u(\ell)$ for $\ell > 2$. Define the random variables $X_1',\ldots,X_a'$ in the same manner as before, and let $X_{12}' = X_1' + X_2'$. Then $J' \equiv X_{12}' + \Sigma_{\ell=3}^{a}\ X_{\ell}'$ and $E(S(J',\ d-i,\ k-1)]$ are interpreted as above, but for the alternative defense instead. $X'_{12}, X'_3, \ldots, X'_a$ are mutually independent, as are $X_{12}, X_3, \ldots, X_a$.

We now show that $X'_{12} \overset{st}{\le} X_{12}$, i.e., X'_{12} is stochastically less than X_{12} [see Barlow and Proschan (1975), p. 110]. X_{12} and X'_{12} have possible values 0, 1 and 2, and

$$P(X_{12}=0) = (1-q_k^{u(1)})\ (1-q_k^{u(2)}) = 1 - q_k^{u(1)} - q_k^{u(2)} + q_k^{u(1)+u(2)} ,$$

$$P(X_{12}=1) = q_k^{u(1)} + q_k^{u(2)} - 2\ q_k^{u(1)+u(2)} ,$$

$$P(X_{12}=2) = q_k^{u(1)+u(2)} .$$

Corresponding expressions hold for the probability distribution of X'_{12}, with $u'(1) = u(1)-1$ and $u'(2) = u(2) + 1$ substituted. Since $u(1) + u(2) = u'(1) + u'(2)$, it follows that $P(X_{12} > 1) = P(X'_{12} > 1)$. Simple algebra yields

$$P(X'_{12} > 0) - P(X_{12} > 0) = q_k^{u(2)}\ (q_k - 1)\ (1 - q_k^{u(1)-u(2)-1}) < 0,$$

and this suffices to show that $X'_{12} \overset{st}{\le} X_{12}$. Since $X'_\ell \overset{st}{=} X_\ell$ for $\ell > 2$, it follows from a simple extension of exercise 1 on page 176 of Barlow and Proschan (1975) that $J' \overset{st}{\le} J$. Since $S(j,\ d-i,\ k-1)$, by virtue of its definition, is a nondecreasing function of j for fixed values of the other arguments, it follows from exercise 15 on page 151 of Barlow and Proschan (1975) that $E[S(J',\ d-i,\ k-1)] \le E[S(J,\ d-i,\ k-1)]$. Thus the alternative defense does not increase the expected damage done to the target, and our verification is now complete.

REFERENCES

[1] BARLOW, R. E. and F. PROSCHAN (1975). *Statistical Theory of Reliability and Life Testing.* Holt, Reinhart and Winston, New York.

[2] BURR, S. A., J. E. FALK and A. F. KARR (1985). Integer Prim-Read Solutions to a Class of Target Defense Problems. *Operations Research, 33,* 726-745.

[3] ECKLER, A. R. and S. A. BURR (1972). *Mathematical Models of Target Coverage and Missile Allocation.* Military Operations Research Society, Alexandria, VA.

[4] FALK, J. E. (1985a). Prim-Read Defense with Defender Shoot-Look-Shoot. Memorandum. Institute for Defense Analyses, Alexandria, VA.

[5] FALK, J. E. (1985b). Minimal Defenses Enforcing Arbitrary Maximum Damage Bounds. *Proceedings of the 1985 IEEE International Conference on Systems, Man and Cybernetics.* Tucson, Arizona, November 12-15, 1985, 651-655.

3. Equation for SEQAT1--Sequential Attack of Unknown Size with One or Two Opportunities to Shoot[1]

There is a single target that may come under an attack by an unknown number of sequentially arriving RV's.

We are protecting this target with D interceptors, and we have sufficient time to observe the results of a first volley against an RV, and fire a second volley, if necessary.

Let

q = probability that a defender targeted against an RV will miss in the first volley,

r = probability that a defender targeted against a surviving RV will miss in the second volley,

s = probability that an RV surviving both volleys will fail to destroy the target, and

p(A,D) = probability that the target protected by D defenders is destroyed by A weapons, given some firing doctrine.

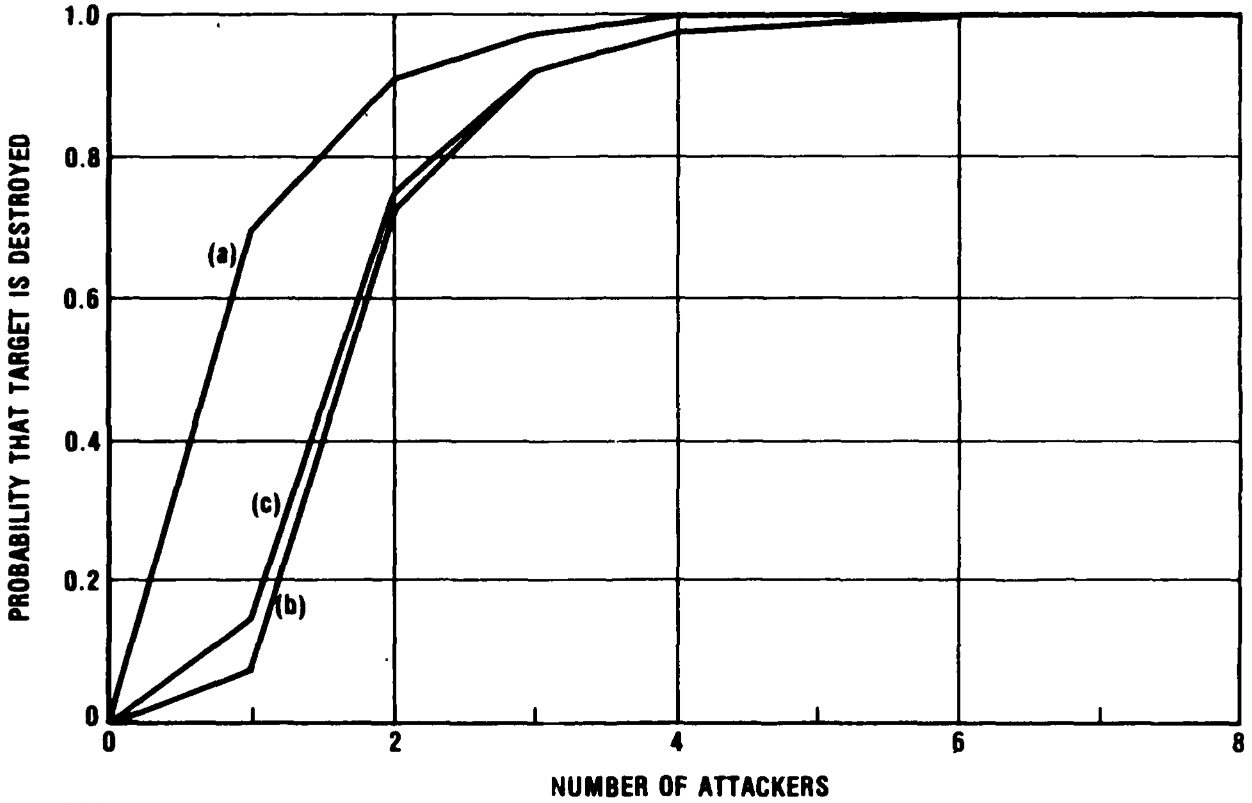
Figure 1 illustrates a pair of possibilities when D=1. Here (q,r,s) = (0.1, 0.2, 0.3), and the results of 3 firing doctrines are displayed. The doctrines are

a) Don't fire at any RV (unprotected case ≡ D=0),

b) Fire one defender in the first volley against the first arriving RV.

c) Fire one defender in the second volley (none in the first) at the first arriving RV.

[1] Source: James E. Falk, "Prim-Read Solutions with Shoot-Look-Shoot", unpublished memorandum, Institute for Defense Analyses, 31 July 1985.



7-11-86-1

Figure 1. Expected Damage Curves for D=1

With only one defender, and with its chances of success being higher if used in the first volley, it is clear that it should be so used. With several defenders, and different q's and r's, the situation becomes less clear.

In keeping with the Prim-Read philosophy, we will use the maximum of the ratios p(A,D)/A (A=1,2,3,...) as the measure of effectiveness of a firing doctrine. In Figure 1, we have

maximum ratio with doctrine (a) = 0.7 $(p_a(1,1)/1)$

maximum ratio with doctrine (b) = 0.361 $(p_b(2,1)/2)$

maximum ratio with doctrine (c) = 0.371 $(p_c(2,1)/2)$

so that firing doctrine (b) is the most effective in that its maximum ratio is the smallest.

In this note, we will assume that the defender will always decide on his first/second volley allocations in such a way as to minimize the maximum of those ratios, where the minimization takes place over all possible such allocations. Under this "behavioral assumption," we make the following definitions.

For each D=0,1,2,..., let

$Q_k(A,D)$ = probability that the target is destroyed by A attackers given that there are D defenders and k of them are sent against the first arriving RV in a first volley, with a second possible, if needed,

$R_k(A,D)$ = probability that the target is destroyed by A attackers given that there are D defenders, the defense has but a single (second) volley at the first arriving RV and sends k defenders at it,

when, in each case, k ranges from 0 to D.

For any given D, set

$$m(D) = \min_{0 \leq k \leq D} \max_{A \varepsilon I^+} Q_k(A,D)/A \tag{1}$$

and let k*(D) denote the smallest integer such that the above minimum is obtained. Given the functions $Q_0(\cdot,D)$, $Q_1(\cdot,D),\ldots,$ $Q_D(\cdot,D)$, both m(D) and k*(D) are well-defined, and k*(D) represents the number of interceptors that the defender fires in his first volley against the first RV that he sees when he has D interceptors left.

In the event that a defender's first volley fails, he is left with a "reserve volley". Assuming that he now has D defenders left, if we have the functions $R_0(\cdot,D)$ $R_1(\cdot,D),\ldots,$ $R_D(\cdot,D)$, we define

$$n(D) = \min_{0 \leq k \leq D} \max_{A \varepsilon I^+} R_k(A,D)/A \tag{2}$$

and let k**(D) denote the smallest integer which minimizes the above. Then k**(D) represents the numbers of interceptors that the defender would fire if he had but one (second) volley to shoot at the first RV that he sees with D interceptors left (he has two volleys at any subsequent RV), and he wishes to choose a damage curve with the smallest slope.

Let

$$Q(A,D) = Q_{k*(D)}(A,D) \tag{3}$$

and

$$R(A,D) = R_{k**(D)}(A,D) \tag{4}$$

For each D, these represent the actual expected damage curves that the defender has selected.

Note that

$$Q(A,0) = R(A,0) = 1-s^A \quad A=0,1,\ldots.$$

(if the attackers are perfect, this holds if we define $0^0=1$).

The following recursions hold

$$R_k(A,D) = r^k(1-s) + (1-r^k(1-s))\, Q(A-1,D-k) \tag{5}$$

$$Q_k(A,D) = q^k R(A,D-k) + (1-q^k)\, Q(A-1,D-k) \tag{6}$$

for $k=0,1,\ldots,D$ and $D\geq 1$. They may be solved in the order:

$$Q(A,0) = R(A,0) \rightarrow R_1(A,1) = R(A,1)$$
$$R(A,1) \rightarrow Q_0(A,1),\ Q_1(A,1) \rightarrow Q(A,1)$$
$$Q(A,1) \rightarrow R_1(A,2),\ R_2(A,2) \rightarrow R(A,2)$$
$$R(A,2) \rightarrow Q_0(A,2),\ Q_1(A,2),\ Q_2(A,2) \rightarrow Q(A,2) \text{ etc.}$$

Note that $R_0(A,D)$ is never computed. The above recursion for $R_0(A,D)$ is

$$R_0(A,D) = (1-s) + s\, Q(A-1,D)$$

and the function $Q(\cdot,D)$ cannot be computed until $R(\cdot,D)$ has been computed. The function $R_0(\cdot,D)$ represents the expected damage when the defender decides <u>not</u> to use his second volley at an unintercepted RV.

In particular

$$R_0(1,D) = 1-s$$

i.e., the probability that the target is destroyed by the first RV when it is not engaged. We now show that $R_0(\cdot,D)$ does not enter in the determination of $R(\cdot,D)$ as long as $r<1$, and therefore need not be computed. The result is intuitive: If you have at least one interceptor left to engage an approaching RV, it is better to use it immediately (even if it has low reliability) instead of saving it for a possible subsequent RV (even if its first-shot reliability might be high).

Lemma. Assume $0\leq r<1$. Then for any $D\geq 1$

$$\frac{Q(A,D)}{A} \leq 1-s \qquad A=1,2,\ldots \tag{7}$$

Proof. Fix $D\geq 1$. We will use induction on A.

Since $Q(0,D) = 0$, we have

$$R_k(1,D) = r^k(1-s) \qquad \text{for any } k\text{, and}$$

so

$$R(1,D) \leq 1-s$$

and

$$Q_k(1,D) = q^k\ R(1,D-k) \leq 1-s$$

and it follows that (7) is true for A=1.

For any k,

$$\begin{aligned}Q_k(A,D) &= q^k\ R(A,D-k) + (1-q^k)\ Q(A-1,D-k)\\ &= q^k[r^k(1-s) + (1-r^k(1-s))\ Q(A-1,D-k)]\\ &\quad + (1-q^k)\ Q(A-1,D-k)\\ &= q^kr^k(1-s) + (1-q^kr^k(1-s))\ Q(A-1,D-k)\\ &\leq q^kr^k(1-s) + (1-q^kr^k(1-s))\ (A-1)(s-1)\end{aligned}$$

where induction is used to get the last inequality. Thus

$$Q_k(A,D) \le [q^k r^k + (1-q^k r^k(1-s)(A-1)](s-1)$$

and the quantity in square brackets is clearly bounded above by the integer A. Thus

$$\frac{Q_k(A,D)}{A} \le 1-s \quad \text{for any } k$$

and the result follows.

Theorem. If $0 \le r,s<1$, then $R_0(\cdot,D)$ need not be used to complete $R(\cdot,D)$.

Proof. To compute $R(\cdot,D)$, we first determine $n(D)$ and $k^{**}(D)$ from (2).

For $k=0$, we have

$$R_0(A,D) = (1-s) + sQ(A-1,D)$$

In particular

$$R_0(1,D) = 1-s$$

and we now show that

$$\frac{R_0(1,D)}{1} \ge \frac{R_0(A,D)}{A} \quad \text{for } A=2,3,... \tag{8}$$

i.e.,

$$A(1-s) \ge (1-s) + s\,Q(A-1,D)$$

i.e.,

$$(A-1) \ge s(A-1 + Q(A-1,D)). \tag{9}$$

But the lemma implies

$$Q(A-1,D) \leq (1-s)(A-1)$$

i.e.,

$$(A-1) \geq s(A-1 + Q(A-1,D))$$

so that (9) and hence (8) is true. Thus

$$\max_{A\varepsilon I^+} \frac{R_0(A,D)}{A} \leq 1-s = R_0(1,D).$$

We will now show that

$$\max_{A\varepsilon I^+} \frac{R_1(A,D)}{A} < 1-s$$

so that k**(D) of (2) is not 0. We need to show that

$$r(1-s) + (1-r(1-s))\ Q(A-1,D-1) < A(1-s) \text{ for all } A\geq 1$$

but, again by the lemma,

$$Q(A-1,D-1) \leq (A-1)(1-s)$$

so that

$$r(1-s) + (1-r(1-s))\ Q(A-1,D-1) \leq r(1-s) + (1-r(1-s))(A-1)(1-s)$$

$$= [r + (1-r(1-s))(A-1)](1-s)$$

and since r<1, the quantity in square brackets is strictly smaller than A and the theorem is proven.

Example. With (q,r,s) = (.1,.2,.3). Table 1 exhibits the values Q(A,D) for A=0,1,...,10 and D=0,1,...,10. The first column of the table (Q(0,A)) represents the probabilities of destruction in the non-defended case.

Table 1. Case (q,r,s) = (.1,.2,.3)

DESTRUCTION PROBABILITIES Q(A,D)

A	Q(A,0)	Q(A,1)	Q(A,2)	Q(A,3)	Q(A,4)	Q(A,5)	Q(A,6)	Q(A,7)	Q(A,8)	Q(A,9)	Q(A,10)
0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	0.700	0.070	0.014	0.014	0.014	0.014	0.014	0.003	0.003	0.003	0.003
2	0.910	0.721	0.137	0.033	0.028	0.028	0.028	0.017	0.007	0.007	0.006
3	0.973	0.916	0.741	0.199	0.055	0.042	0.041	0.031	0.021	0.012	0.010
4	0.992	0.975	0.922	0.760	0.257	0.081	0.056	0.045	0.034	0.025	0.016
5	0.998	0.992	0.977	0.928	0.777	0.311	0.109	0.079	0.052	0.039	0.030
6	0.999	0.998	0.993	0.978	0.933	0.793	0.361	0.176	0.104	0.060	0.046
7	1.000	0.999	0.998	0.994	0.980	0.938	0.808	0.418	0.239	0.131	0.074
8	1.000	1.000	0.999	0.998	0.994	0.981	0.942	0.825	0.470	0.296	0.162
9	1.000	1.000	1.000	0.999	0.998	0.994	0.983	0.948	0.841	0.518	0.349
10	1.000	1.000	1.000	1.000	0.999	0.998	0.995	0.984	0.952	0.855	0.561

THE NUMBER TO SHOOT AT NEXT RV IN 1ST VOLLEY:

0	1	1	1	1	1	1	1	1	1	1

IF MISS, THE NUMBER TO SHOOT IN 2ND VOLLEY:

0	0	1	1	1	1	1	2	2	2	2

SLOPES(D):-

0.700	0.361	0.247	0.190	0.156	0.134	0.118	0.105	0.095	0.086	0.056

Also included in Table 1 is the firing doctrine which yields the Q(A,D) values. To incorporate this doctrine, suppose, for example, that the defender has 10 interceptors at the target. If he sees an attacker coming in, he uses one of these 10 in the first volley and, if he misses, he shoots two in the second volley. These values are read below column Q(A,10) of the Q(A,D) table. If the first volley is unsuccessful, but the second succeeds, the defender now has 10-1-2 = 7 interceptors left to deal with any subsequent attackers. If indeed, a second RV attacks, the defender again uses a 1-2 firing doctrine as indicated below column D=7. If, again, his first volley fails but his second succeeds, he has 7-1-2 = 4 interceptors left.

A possible (but unlikely*) battle history with D=10 is:

RV #	1st Volley	2nd Volley	# Interceptors Left if 1st Volley Misses
1	1	2	7
2	1	2	4
3	1	1	2
4	1	1	0
5	.	.	.

The slopes m(D) of the lowest maximum damage curves which yield an upper bound on the expected damage are also given in Table 1. Thus, for example, with 7 interceptors, the expected damage is bounded above by the linear function F(A) = 0.105A.

Figure 2 exhibits the expected damage curves for increasing values of D.

Often one wishes to determine the minimum number of interceptors required to enforce a maximum damage function of a given slope. Figure 3 exhibits the curves both with and without an SLS capability when the interceptor failure probabilities are 0.3 and the RV failure probability is 0.1. For example, if one wishes to design a defense whose expected damage per attacking

*The probability that this would occur is

$$(.1)(1-.2^2)(.1)(1-.2^2)(.1)(1-.2)(.1)(1-.2) = 5.9 \times 10^{-5}$$

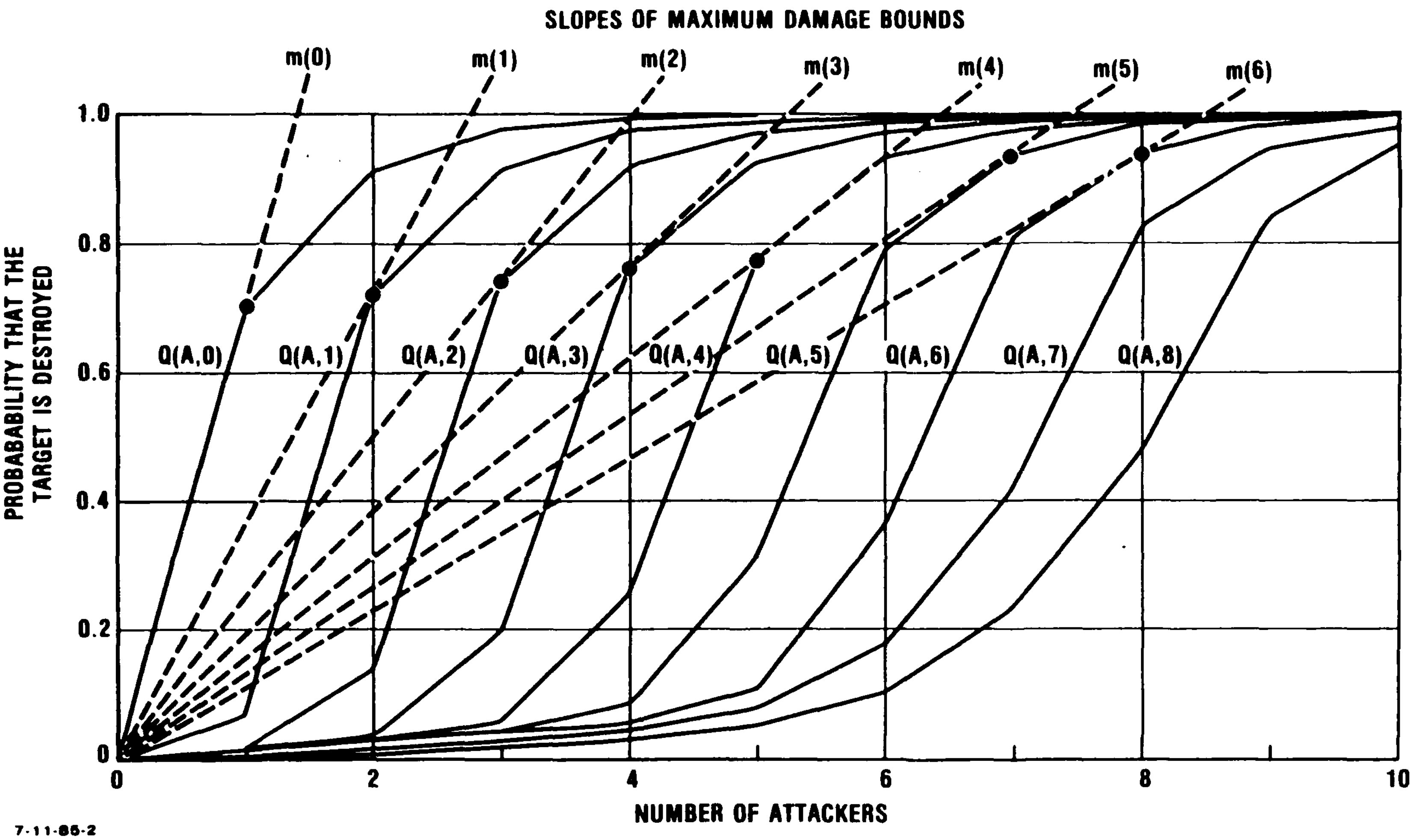


Figure 2. Expected Damage Curves and Maximum Damage Bounds for Varying D

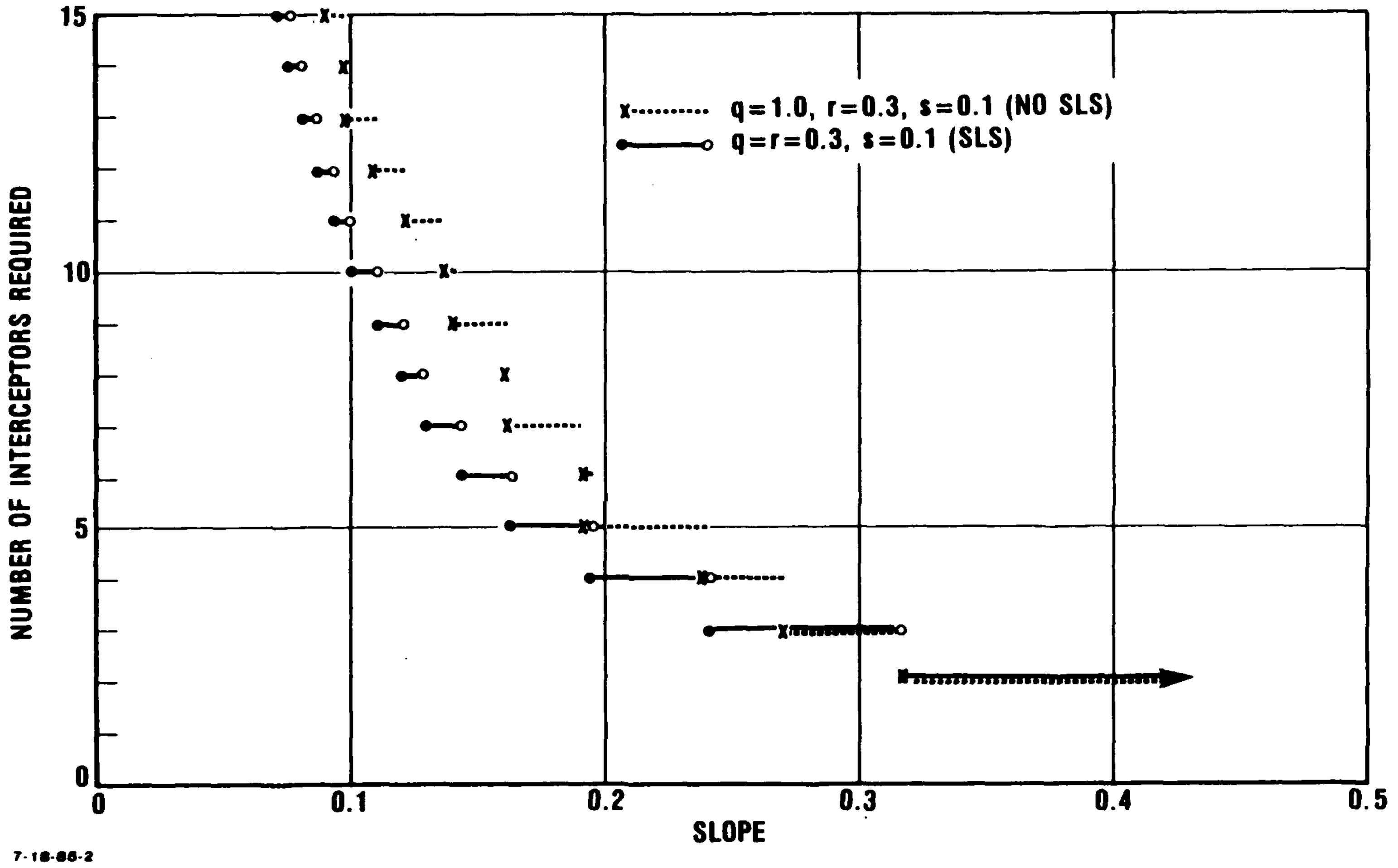


Figure 3. Number of Interceptors Required for a Given Slope

weapon is 0.15, one needs 6 interceptors with an SLS capability, and 9 interceptors without.

Extensions and Implementation

The case q=1 corresponds to no SLS capability. The case s=1 is a trivial case wherein the RV's are completely unreliable so no defense is needed.

The results of this note are easily extended to allow for additional volleys.

A FORTRAN program has been coded to generate the values Q(A,D) for any range of A,D values.

4. Equation for SEQAT2--Sequential Attack of Known Size[1]

Here we address the case where the attack is "sequential", i.e., there is enough time between successive attackers that they can be ordered and the attack size is known. We define, as before:

P(A,D) = probability that the target survives, given that it is under attack by A missiles and is optimally defended by D defenders.

Obviously, if the defender knows the value of A, he will defend uniformly according to the result of Appendix A. (A simultaneous attack can be considered sequential by numbering the attackers in any order.)

However, if the defender has a shoot-look-shoot capability, and sufficient time between arrivals, he can choose to structure his defense in volleys, with the prospect of saving defenders for use against future attackers.

Suppose the defender has time for two volleys against each incoming attacker. Let a be, as before, the kill probability of an attacking missile. Let

d = probability that a defending interceptor will destroy an attacking missile in the first volley

and

e = probability that a defending interceptor will destroy an attacking missile in the second volley.

Let

d(A) = number of interceptors to shoot at the first of A attacking missiles in the first volley

and

[1] Source: Appendix C of Reference [1] on page R-1.

$e(A)$ = number of interceptors to shoot at the first of A attacking missiles in the second volley, given that the first volley has failed.

Then

$1-d(1-d)^{d(A)}$ is the probability that the first volley is successful,

$(1-d)^{d(A)}(1-(1-e)^{e(A)})$ is the probability that the first volley fails but the second is successful,

and

$(1-d)^{d(A)}(1-e)^{e(A)}(1-a)$ is the probability that both volleys fail and the attack also fails.

The following recursion holds:

$$P(A,D) = \max_{\substack{d(A),e(A)\ \epsilon\ I^t \\ d(A) + e(A) \le D}} \left\{ (1-(1-d)^{d(A)})\ P(A-1,D-d(A)) \right. \\ \left. + (1-d)^{d(A)}(1-a(1-e)^{e(A)})P(A-1,D-d(A)-e(\right.$$

with

$$P(0,D) = 1 \quad \text{for all } D\ \epsilon\ I^+.$$

Given a, d and e, the recursion can be solved by dynamic programming to determine $P(A,D) = p_{ij}$. Note that the solution of this recursion would agree with the results of Appendix A in the case where e=0.

Obviously, the above recursion could be extended if the defender had more than two opportunities to protect himself.

DISTRIBUTION

IDA PAPER P-1875

ROBUST PREALLOCATED PREFERENTIAL DEFENSE MODEL

30 Copies

Copies

Director,
Office of the Joint Chiefs of Staff
Washington, D.C. 20301-5000
ATTN: Director, Joint Analysis Directorate 1

Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia

Deputy Chief of Staff for Research and Development
and Acquisition
Department of the Army
Room 3A474, The Pentagon
Washington, D.C. 20310
ATTN: Director, Missile and Air Systems Division
RDA/DAMA-WS 1

Office of the Chief of Staff
Department of the Army
Ballistic Missile Defense Program Office
P.O. Box 15280
Arlington, Virginia 22215
ATTN: DACS-BMZ 1

Commander
Department of the Army
Ballistic Missile Defense Systems Command
P.O. Box 1500
Huntsville, Alabama 35807
ATTN: Library 1

Headquarters
Department of the Air Force
Assistant Chief of Staff Studies and Analysis
Room 1E388, The Pentagon
Washington, D.C. 20330
ATTN: Library 1

Directorate of Aerospace Studies
Deputy Chief of Staff, Plans and Programs
Headquarters, Air Force Systems Command
Kirtland AFB, NM 89117
ATTN: Library 1

The Rand Corporation
P.O. Box 2138
Santa Monica, California 90406-2138
ATTN: Library 1

The Rand Corporation
2100 M Street, N.W.,
Washington, D.C. 20037
ATTN: Library 1

Hudson Institute, Inc.
Center for Naval Analysis (CNA)
P.O. Box 11280
Alexandria, Virginia 22311
ATTN: Library 1

Los Alamos National Laboratory
P.O. Box 1663, Mail Station 5000
Los Alamos, NW 87545
ATTN: Library 1

University of California
Lawrence Livermore National Laboratory
P.O. Box 808
Livermore, CA. 94550
ATTN: Library 1

Analytic Services, Inc. (ANSER)
Crystal Gateway 3
1215 Jefferson Davis Highway
Arlington, Virginia 22202
ATTN: Library 1

Teledyne-Brown Engineering
Cummings Research Park
Huntsville, Alabama 35807
ATTN: Library 1

Teledyne-Brown Engineering
1250 Academy Park Loop
Colorado Springs, Colorado 80910
ATTN: Library 1

SAIC
1710 Goodridge Drive
McLean, Virginia 22102
ATTN: Library 1

McDonnell-Douglas Astronautics Company
5301 Bolsa Avenue
Huntington Beach, CA. 92647
ATTN: Library 1

Sparta, Inc.
4901 Corporate Drive, Suite 102
Huntsville, Alabama 35805
ATTN: Library 1

System Planning Corporation
1500 Wilson Boulevard
Arlington, Virginia 22209
ATTN: Library 1

Institute for Defense Analyses
1801 North Beauregard STreet
Alexandria, Virginia 22311
ATTN: 10

Dr. J. Bracken	2
Dr. W.J. Schultis	1
Mr. R.B. Pirie	1
Mr. S.J. Deitchman	1
Control & Distribution	5